



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA PODNIKATELSKÁ  
ÚSTAV INFORMATIKY**

FACULTY OF BUSINESS AND MANAGEMENT  
INSTITUTE OF INFORMATICS

# **ERP SYSTÉMY - KOMPRESSE SQL DATABÁZE**

ERP SYSTEMS - SQL DATABASE COMPRESSION

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**LUKÁŠ FUXA**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. JIŘÍ KŘÍŽ, Ph.D.**

BRNO 2014





## **Abstrakt**

Předmětem této bakalářské práce je návrh na kompresi SQL databáze u informačního systému v nadnárodní společnosti. Výstupem mé práce bude návrh na kompresi SQL databáze. Firma si stanovila určité požadavky, které musí být dodrženy. Na základě těchto požadavků navrhnu různá řešení a z nich nakonec vyberu jedno, dle mého názoru nejlepší, které na daný problém aplikuji.

## **Abstract**

Subject of this thesis is proposal to compression of SQL database at information system used in multinational company. The aim of my work will be proposal of SQL database compression. The company particularizes requirements, which have to be kept. On the basis of this requirements I will propose various solutions and from these solutions will be choose one, in my opinion the best, which will be implemented.

## **Klíčová slova**

Informační systém, SQL databáze, IS, Kompresí SQL databáze, ERP systém

## **Keywords**

Information system, SQL database, IS, SQL database compression, ERP system

## **Bibliografická citace**

FUXA, L. ERP systémy - komprese SQL databáze. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2014. 59 s. Vedoucí bakalářské práce Ing. Jiří Kříž, Ph.D..

## **Čestné prohlášení**

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně.  
Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne .....

.....

podpis

## **Poděkování**

Tímto bych rád poděkoval Ing. Jiřímu Křížovi, Ph.D. za jeho rady a připomínky při psaní této bakalářské práce. A dále také paní Ing. Veronice Navrátilové za oponenturu.

# Obsah

ÚVOD.....	10
CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ.....	11
1 TEORETICKÁ VÝCHODISKA PRÁCE .....	12
1.1 Informační systém .....	12
1.1.1 Druhy IS.....	12
1.2 ERP systém.....	16
1.2.1 Trendy v ERP systémech.....	17
1.2.2 Klasifikace ERP systémů.....	19
1.3 Databáze .....	19
1.3.1 Relační model .....	21
1.3.2 Systém přístupu k databázi .....	21
1.4 Jazyk SQL .....	21
1.5 Databázová platforma SQL server .....	22
1.5.1 Fyzické soubory databáze .....	22
1.5.2 Komprese SQL databáze .....	24
1.5.3 Porovnání Microsoft SQL serveru 2005 a Microsoft SQL serveru 2008 ...	24
1.6 Počítačový Server .....	25
1.6.1 Mikroprocesor.....	25
1.6.2 Operační paměť .....	25
1.6.3 Pevné disky a disková pole.....	26
2 ANALÝZA SOUČASNÉHO STAVU .....	30
2.1 O společnosti .....	30
2.2 Hardware .....	30
2.2.1 Datové centrum Nagano .....	31
2.3 Software.....	33
2.4 Databáze .....	33
2.4.1 Obsah databáze .....	33
2.4.2 Noční a víkendový provoz databáze.....	34



2.4.3	Aktuální vývoj a stav databáze .....	34
2.5	Požadavky zákazníka.....	36
3	VLASTNÍ NÁVRHY ŘEŠENÍ.....	38
3.1	Online komprese.....	38
3.1.1	Výhody online komprese .....	40
3.1.2	Nevýhody online komprese .....	40
3.2	Jednorázový odsun dat .....	41
3.2.1	Výhody jednorázového odsunu .....	44
3.2.2	Nevýhody jednorázového odsunu.....	44
3.3	Start nové databáze.....	45
3.3.1	Výhody startu nové databáze .....	45
3.3.2	Nevýhody startu nové databáze .....	48
3.4	Výběr optimálního řešení .....	49
3.5	Aplikace vybraného řešení .....	50
	ZÁVĚR .....	53
	SEZNAM POUŽITÉ LITERATURY .....	54
	SEZNAM POUŽITÝCH ZKRATEK.....	56
	SEZNAM OBRÁZKŮ.....	57
	SEZNAM GRAFŮ .....	58
	SEZNAM TABULEK .....	59

## ÚVOD

V dnešní době počítačů se většina podniků neobejde bez fungujícího podnikového informačního systému, který bude zpracovávat a uchovávat veškerá provozní data společnosti. Kvalitní informační systém dokáže velmi zefektivnit různé procesy v podniku, který tím může zvýšit své úspěchy na dnešních přeplněných trzích. Součástí každého podnikového informačního systému jsou databáze, o které se musíme pravidelně starat. Mezi nástroje pro údržbu databáze patří například zálohování, kontrola integrity, aktualizace, komprese a jiné. Já se ve své bakalářské práci budu především věnovat kompresi, kterou využiji k tvorbě návrhu na zmenšení velikosti databáze v konkrétní vybrané společnosti.

Tato databáze se dostala do stavu, kdy s ní nebylo téměř možné vůbec pracovat. Mnou vybraná společnost, která si nepřála být jmenována, využívá platformu Microsoft SQL server 2005, na níž je databáze spuštěna.

## **CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ**

Cílem mé práce je návrh na kompresi SQL databáze, díky kterému se zrychlí provádění veškerých operací, jako jsou například přesun, zálohování, výpočet indexů atd.

Zaměřím se na analýzu současného stavu, kde popíši mnou vybranou společnost a stav, ve kterém se databáze nachází. Dalším krokem bude návrh možných řešení komprese databáze tak, aby byly splněny zadané podmínky od společnosti. Mnou navržené řešení je univerzální, tudíž se dá použít na jakýkoli podnikový informační systém, který ke své práci využívá SQL databázi.

# 1 TEORETICKÁ VÝCHODISKA PRÁCE

V této části se zaměřím na teoretické poznatky, které budou v další části práce převedeny do praktické části.

## 1.1 Informační systém

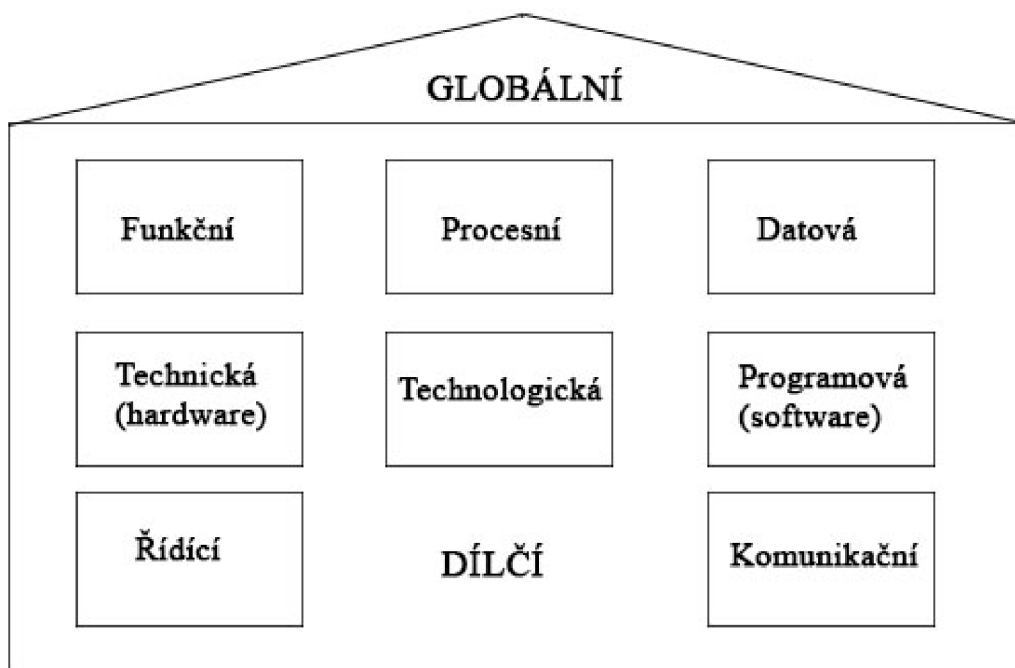
Informační systém (dále jen IS) je v dnešní době základem, pro každou rozvíjející se organizaci, která má v záměru strategické zpracování informací. Klíčem k dosažení tohoto cíle je právě IS [1].

*„IS vytvářejí lidé, kteří prostřednictvím dostupných technologických prostředků a stanovené metodiky zpracovávají podniková data a vytvářejí z nich informační a znalostní bázi organizace sloužící k řízení podnikových procesů, manažerskému rozhodování a správě podnikové agendy [1 str. 61].“*

### 1.1.1 Druhy IS

IS lze klasifikovat ze čtyř různých pohledů.

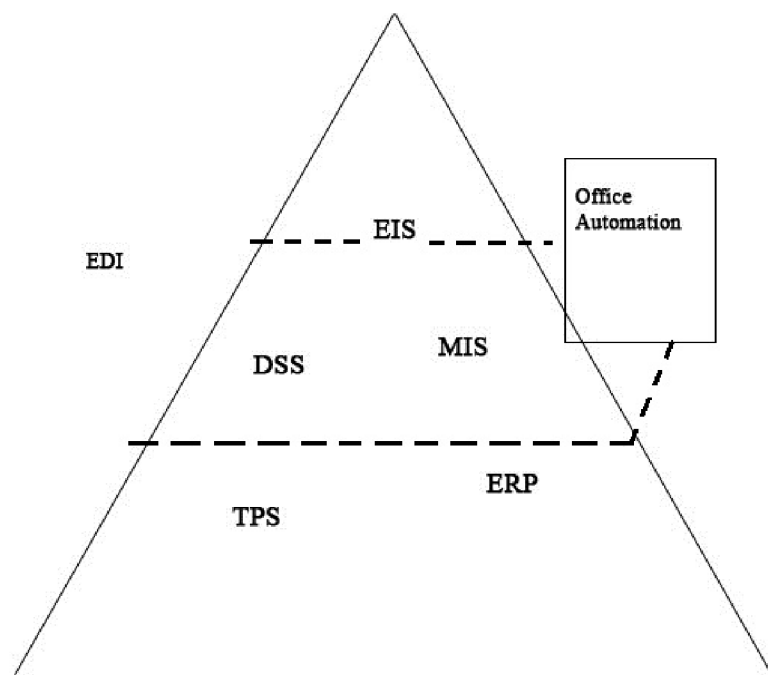
Jako první bych uvedl klasifikaci z pohledu architektury.



Obrázek č. 1: IS z pohledu architektury [2]

- Globální architektura je základním schématem IS. Tato architektura je tvořena jednotlivými dílčími částmi, které popisují návrhy IS dle různých hledisek – lze zde vidět analogii s plány rozvodu vody, elektřiny a plynu v domě [2].
- Funkční architektura dělí IS na menší podsystémy, skupiny funkcí (např. mzdy, studenti) pomocí dekompozice globální architektury [2].
- Procesní architektura je zaměřena na popis budoucího stavu procesů, které jsou zaměřeny na neautomatizované činnosti a funkce IS. Hlavním cílem této architektury je určit vazby podniku s okolím [2].
- Technická (hardwarová) architektura určuje rozmístění výpočetních a komunikačních technologií. Obsahuje specifikace počítačových sítí, serverů a dalších zařízení [2].
- Technologická architektura má za úkol zpracování jednotlivých aplikací v závislosti na definované technické, datové či programové architektuře. Součástí technologické architektury je způsob zpracování aplikací, způsob zpracování dat, vnitřní stavba aplikací a uživatelské rozhraní aplikací [2].
- Datová architektura vychází z vhodného datového modelu a představuje návrh datové základny organizace. Výsledkem datové architektury může být E-R diagram, který navrhuje datové entity, vazby a atributy [2].
- Programová (softwarová) architektura určuje jaké programy či programové komponenty bude IS obsahovat [2].
- Komunikační architektura určuje vnější rozhraní IS a jeho komunikaci s okolím.
- Řídící architektura definuje pravidla, která určují jak má IS fungovat [2].

Dále se lze na IS dívat z pohledu úrovně řízení.

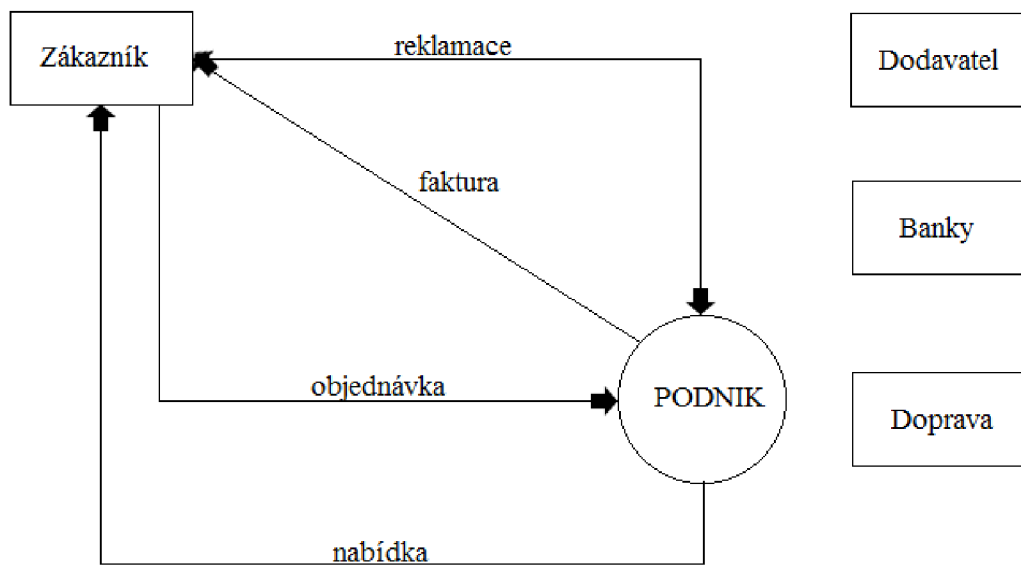


Obrázek č. 2: IS z pohledu úrovně řízení [2]

- ERP (Enterprise Resource Planning) pokrývají kompletní problematiku procesů [2].
- TPS (Transaction Processing Systems) jsou nástupci agend umístěných u pracovníka např. „Objednávka zboží“ [2].
- MIS (Management Information Systems) jsou určeny k taktickému řízení, provádějí sumarizaci a agregaci dat za určité období [2].
- DSS (Decision Support Systems) jsou systémy, které napomáhají při rozhodování. Spadají sem většinou analýzy z MIS, které jsou určené pro taktické i strategické řízení [2].
- EIS (Executive Information Systems) obsahují systémy určené pro vrcholové vedení. Tyto systémy umožňují přístup k externím datům [2].

- Office Automation je automatizace administrativy. Obsahem této části jsou veškeré textové editory, kalendáře či elektronická pošta [2].
- EDI (Electronic Data Interchange) je část IS, která se zabývá komunikací podniku s jeho okolím [2].

Předposlední klasifikací je pohled z okolí.



Obrázek č. 3: IS z pohledu okolí [2]

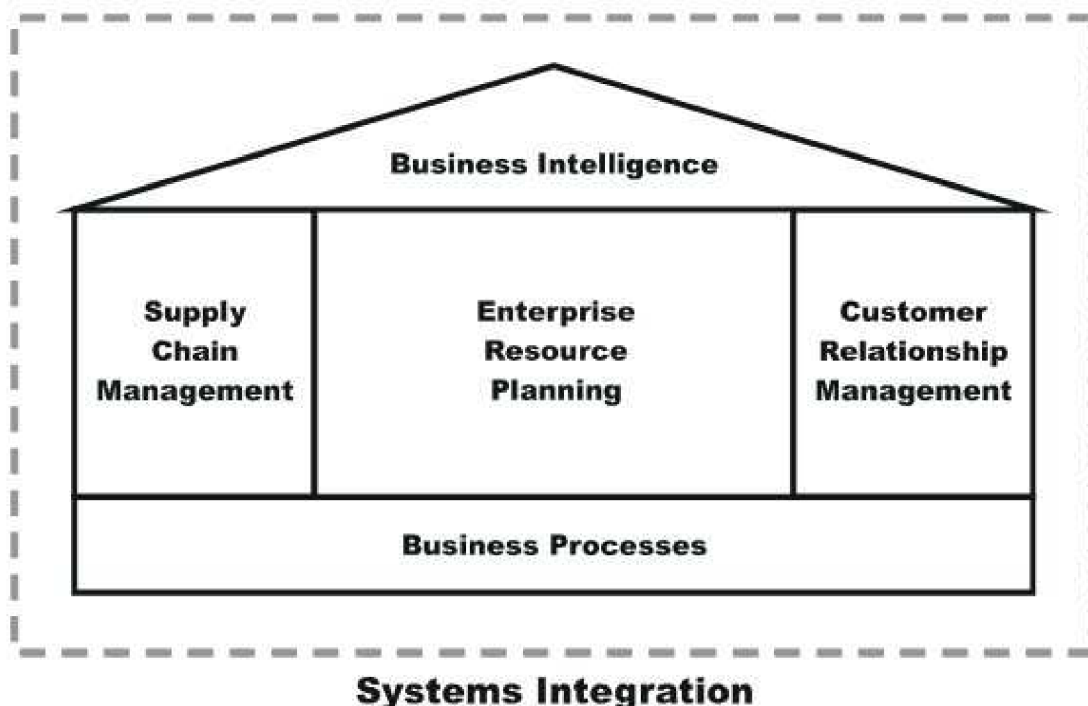
Při tomto pohledu, který je obvykle zachycen kontextovým diagramem, jsou sledovány klíčové toky dat a úlohy vně podniku.

Poslední je holisticko-procesní klasifikace, která se skládá ze čtyř hlavních částí. Tyto části jsou:

- ERP jádro, které je zaměřené na řízení podnikových procesů [1].
- CRM (Customer Relationship Management) systém, který se stará o procesy směřující k zákazníkům [1].
- SCM (Supply Chain Management) systém, který řídí dodavatelský řetězec, jehož součástí bývá systém sloužící k plánování a rozvrhování výroby [1].

- BI (Business Intelligence) sbírá data z ERP, CRM a SCM systémů a na základě získaných dat poskytuje informace pro rozhodovací procesy podnikového managementu [1].

Systémová integrace poskytuje prostředky, díky kterým lze vytvářet a udržovat podnikový informační systém na technologické, řídicí, projektové i strategické úrovni [1].



Obrázek č. 4: Holisticko-procesní pohled na podnikové informační systémy [3]

## 1.2 ERP systém

Při svém vzniku bylo na ERP systémy nahlíženo s jistou nedůvěrou, avšak v dnešní době jsou nedílnou součástí velkých a středních společností, a dokonce už nacházejí využití i v menších podnicích. ERP lze definovat jako informační systémy, pomocí kterých lze řešit plánování a řízení všech klíčových podnikových procesů, a to na všech úrovních podnikové architektury. ERP systémy jsou určeny také k tomu, aby v těchto klíčových procesech podniku zvýšily efektivitu. Mezi klíčové procesy lze zahrnout např. logistiku, výrobu, lidské zdroje, zakázkové zpracování či finanční analýzy spolu s ekonomikou [4].



Avšak ne všechny dostupné podnikové informační systémy zaměřené na řízení podniků v oblasti správy ekonomiky, lidských zdrojů, logistiky a výroby se dají označovat jako ERP systémy. Spousta z nich postrádá základní vlastnosti ERP systémů jako je potřebný rozsah, hloubka funkcionality či míra technologické vyspělosti, takové systémy se označují jako ekonomické a jsou zaměřené primárně na účetnictví [1].



Obrázek č. 5: Struktura ERP systému [5]

Jak již bylo řečeno, tak ke klíčovým procesům ERP systémů patří:

- výroba,
- nákupní, prodejní a výrobní logistika,
- lidské zdroje,
- ekonomika [1].

Mezi hlavní požadavky, které jsou kladené na ERP systémy, lze zařadit realizaci měřitelných přínosů v oblasti snižování nákladů, které vznikají neefektivním řízením

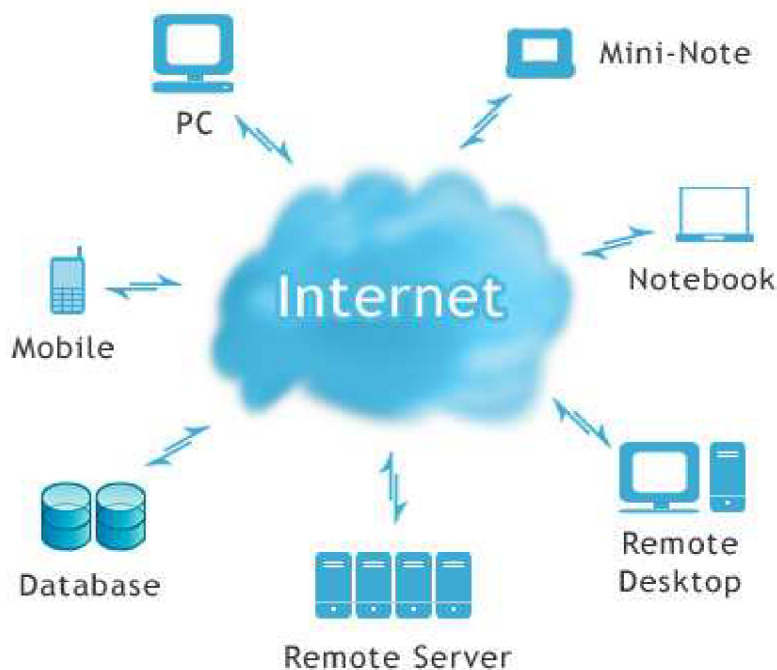
společnosti, nebo realizaci neměřitelných přínosů v oblasti řízení podnikových procesů a dostupnosti v reálném čase [1].

Každý ERP systém by měl splňovat pět základních vlastností a to:

- Automatizaci a integraci hlavních podnikových procesů [1].
- Sdílení dat, postupů a jejich standardizaci přes celý podnik [1].
- Vytváření a zpřístupňování informací v reálném čase [1].
- Schopnost zpracovávat historická data [1].
- Celostní přístup k prosazování ERP koncepce, díky kterému lze odpovědět na otázky jak a s kým provést implementaci, jak a v čem školit uživatele, jaké pracovní nároky je potřeba změnit atd. [1].

### 1.2.1 Trendy v ERP systémech

V dnešní době se v této oblasti rozšiřují spíše trendy technologické. Spousta společností si nemůže dovolit vlastní systém nebo nemá dostatečnou kapacitu se o něj starat, proto se často hovoří o propojení datových služeb s výpočetními cloudy [5].



Obrázek č. 6: SaaS [5]

Na předchozím obrázku je uvedena technologie SaaS (Software as a Service), což označuje software jako službu, kde dochází k nabídce služby provozovatelem přes Internet. Tento model umožňuje uživatelům přistupovat ke svým datům odkudkoli a z jakéhokoli zařízení, které je připojené do sítě internet [5].

### 1.2.2 Klasifikace ERP systémů

Rozdělení ERP systémů je detailně popsáno v následující tabulce.

Tabulka č. 1: Klasifikace ERP systémů [1]

ERP systém	Charakteristika	Výhody	Nevýhody
<b>All-in-One</b>	Schopnost pokrýt všechny klíčové interní podnikové procesy (personalistka, výroba, logistika, ekonomika)	Vysoká úroveň integrace, dostačující pro většinu organizací	Nižší detailní funkcionalita, nákladná customizace
<b>Best-of-Breed</b>	Orientace na specifické procesy nebo obory, nemusí pokrývat všechny klíčové procesy	Špičková detailní funkcionalita, nebo specifická oborová řešení	Obtížnější koordinace procesů, nekonzistence v informacích, nutnost řešení více IT projektů
<b>Lite ERP</b>	Odlehčená verze standartního ERP zaměřena na trh malých a středně velkých firem	Nižší cena, orientace na rychlou implementaci	Omezení ve funkcionalitě, počtu uživatelů, možnostech rozšíření atd.

- All-in-One systémy jsou schopny pokrýt a integrovat všechny čtyři klíčové procesy ERP systémů vyjmenované výše. Většina řešení, které jsou nabízeny na mezinárodních trzích, však nepokrývá jeden z klíčových procesů a to personalistiku. Při implementaci bývá tento proces nahrazen subdodávkou od lokálního dodavatele, který se na tento proces specializuje. Dodavatelé těchto systémů, které nepokrývají proces personalistiky, obvykle garantují dodávku kompletního systému včetně dané subdodávky a integrace. Volba All-in-One ERP systémů pro podniky by měla znamenat realizaci pouze jednoho implementačního projektu [1].

- Best-of-Breed nemusejí nutně pokrývat všechny čtyři procesy, ale umí zákazníkovi poskytnout detailní funkcionalitu nebo jsou orientované na určité obory podnikání. Tyto systémy jsou nasazovány buď samostatně, nebo společně s jinými informačními systémy tvoří jakousi podnikovou ERP koncepci [1].
- Lite ERP systémy tvoří specifickou nabídku, která se vyznačuje nižší cenou a různými omezeními [1].

### 1.3 Databáze

V dnešní době je celá společnost postavena na databázových systémech. Databáze můžeme vidět ve všemožných pracovních odvětvích, jakou jsou výzkumy, zdravotnictví, školství atd. Pod pojmem databáze si lze představit soubor dat, který slouží k popisu reálného světa. V databázích se používají odborná označení entita a atribut. Entita označuje prvek reálného světa (např. člověk), kdežto atributy nám udávají vlastnosti entit (např. u člověka to může být jméno, datum narození). Standardem dnešní doby je tzv. relační model, který popíše v následující kapitole [6].

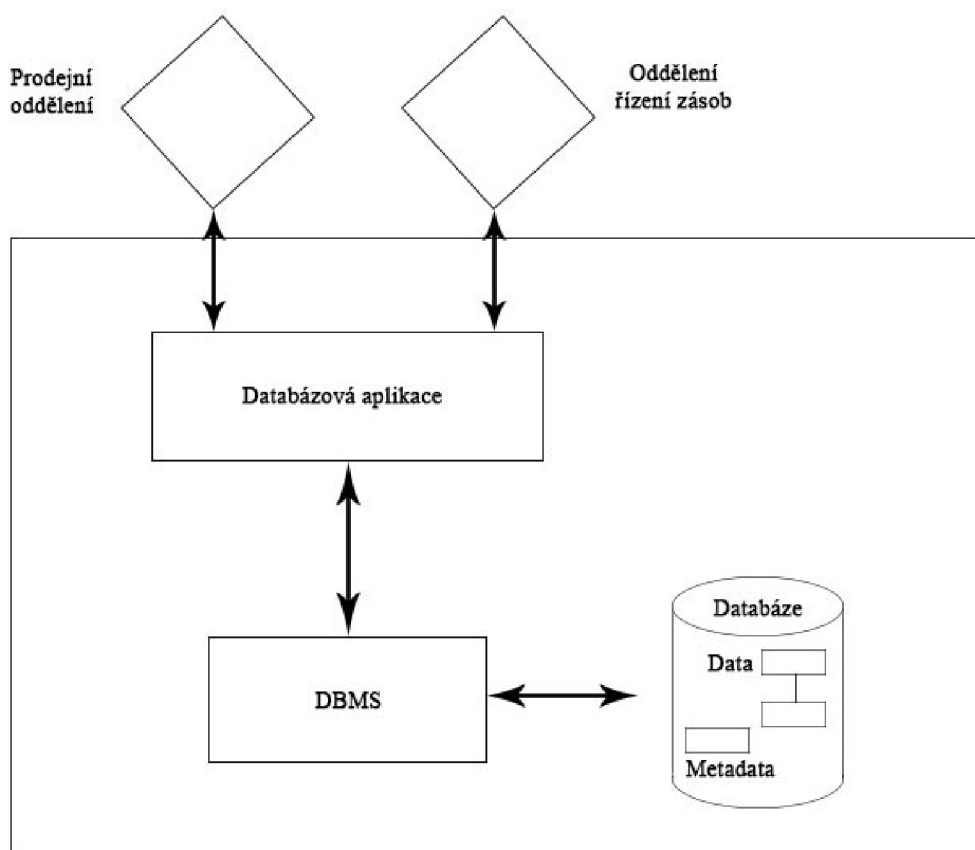
V následujících větách se pokusím co nejvíce objasnit pojem databáze. Databáze je jediné, veliké úložiště dat. Tato úložiště může využívat více oddělení či uživatelů zároveň. Veškerá využívaná data jsou integrována s minimálním množstvím duplikací. Podstatná věc je, že databázi jako takovou, nevlastní žádné oddělení či uživatel, ale je sdíleným obsahem společnosti [7].

Databáze však neobsahuje pouze provozní data, ale obsahuje také popis těchto dat, proto je databáze často označována jako sebepopisující kolekce integrovaných záznamů. Popis těchto dat, který se označuje jako metadata, tvoří slovník dat. Tento charakter databáze způsobuje, že data jsou nezávislá. To znamená, že pokud se do existující databáze přidávají nové struktury nebo upravují stávající struktury, tak aplikace využívající databázi zůstávají nezměněny, pokud přímo nesouvisí s provedenými změnami [7].

### 1.3.1 Relační model

Základním pojmem tohoto modelu je relace. Relaci lze definovat jako tabulku skládající se ze sloupců a řádků. Sloupce v těchto tabulkách odpovídají jednotlivým vlastnostem (atributům) entity, zatímco řádky zobrazují aktuální stav světa. Tabulka je základem pro tvorbu kompletní databáze. Zjednodušeně řečeno relace odpovídá tabulce, která obsahuje prvky relace. Tyto prvky odpovídají konkrétnímu řádku v tabulce. Jeden řádek tabulky lze označit jako databázový záznam. Soubor tabulek (relací) tvoří kompletní databázi (relační schéma) [6].

### 1.3.2 Systém přístupu k databázi



Obrázek č. 7: Systém přístupu k databázi [7]

Předchozí obrázek znázorňuje systém přístupu k databázím. Konkrétně ukazuje, jak prodejní oddělení a oddělení řízení zásob využívají své aplikační programy pro přístup k databázi pomocí DBMS (Database Management System). Každá množina

databázových aplikací zpracovává datové položky, údržbu dat a generování zpráv, zatímco DBMS spravuje fyzickou strukturu a uložení databáze [7].

System řízení databáze neboli DBMS je systém umožňující definovat, vytvářet a udržovat databázi. Nadále také poskytuje řízený přístup k této databázi. DBMS umožňuje uživatelům vkládat, aktualizovat, mazat a vyvolávat data z databáze. Jestliže máme existující databázi, která obsahuje data i popis dat, tak nám DBMS může poskytnout možnost dotazování se na tato data, což se nazývá dotazovací jazyk. Mezi tyto jazyky patří např. SQL [7].

Databázová aplikace je software, který pracuje s databází vyvoláním odpovídajícího požadavku pro DBMS. Uživatelé pracují s databází prostřednictvím určité množiny databázových aplikací. Tyto aplikace se používají k vytváření a správě databáze a ke generování informací, a jsou tvořeny obvyklými dávkovými aplikacemi nebo, což je dnes využívanější, online aplikacemi [7].

## **1.4 Jazyk SQL**

Historie jazyka SQL spadá do 70. a 80. let minulého století. Zkratka SQL (Structured Query Language) v překladu znamená strukturovaný dotazovací jazyk. Díky tomuto jazyku lze vytvářet databáze a dále s nimi manipulovat. Tento jazyk spadá pod tzv. deklarativní programovací jazyky, což v praxi znamená, že kód se nepíše v samostatném programu, ale je vložen do jiného programovacího procedurálního jazyka [6].

## **1.5 Databázová platforma SQL server**

Pod platformou SQL server se neskrývá pouze databáze, ale komplexní, výkonná, moderní, spolehlivá a hlavně bezpečná serverová platforma, díky které můžeme ukládat a spravovat údaje v databázích či datových skladech. SQL server nalezne uplatnění v široké škále podniků všech velikostí. V dnešní době existuje více dodavatelů SQL serveru, mezi nejznámější patří Microsoft či Oracle. Já se ve své práci budu zabývat SQL serverem od společnosti Microsoft [8].

Tato platforma v sobě sdružuje několik vzájemně se doplňujících nástrojů. Mezi tyto nástroje patří:

- Databáze - pod tímto pojmem si lze představit cokoliv od kartotéky u lékaře po údaje organizované databázovým serverem. Obecně řečeno databáze obsahuje údaje a nástroje pro jejich ukládání, manipulaci a správu [8].
- Databázový server, pod kterým si lze představit soubor softwarových prostředků, díky kterým můžeme pracovat s uloženými údaji, ale i organizovat a realizovat přístupy klientů k těmto údajům [8].
- Soubor nástrojů pro správu a zabezpečení údajů v databázi [8].

### **1.5.1 Fyzické soubory databáze**

Microsoft SQL server zajišťuje různé funkce prováděné s daty, jako jsou transakce, vazby, indexování a mnoho dalších. Databáze se skládá vždy ze dvou typů souborů a to z databázového souboru (soubor MDF) a z transakčního logu (soubor LDF) [9].

Přestože pro správné fungování databáze jsou nutné oba typy souborů, tak důležitější je přece jen datový soubor. Tento soubor má za úkol uchovávat všechny informace o aktuálním stavu dat v databázi. Dá se říci, že uchovává prakticky vše, co databázi definuje strukturou i daty. Pokud je provedena jakákoli změna v datech či struktuře, tak se tato změna zároveň projeví i v datovém souboru [9].

Transakční log má za úkol uchování historie provedených databázových operací. Hlavním úkolem je zajištění nedělitelnosti operací, což znamená, že se buď provede celá operace, nebo se operace neprovede vůbec. Nemůže se provést pouze část operace [9].

Díky transakčnímu logu je Microsoft SQL server schopen při nečekaném výpadku všechny dokončené transakce, které nebyly správně uloženy v datovém souboru, spustit znovu, anebo všechny nedokončené operace, které však byly uloženy do datového souboru, vrátit zpět [10].

Transakční log je rozdělen do několika částí, které se označují Virtual log files (VLFs). Počet a velikost těchto souborů je ovlivněn velikostí růstu logu. Každý soubor v databázi, ať už se jedná o datový soubor či transakční log, může mít nastavenou možnost automatického zvětšení (autogrow) [10].

**Tabulka č. 2: Počet a velikost VLFs v závislosti na zvětšení transakčního logu [10]**

Zvětšení transakčního logu	Množství VLFs	Velikost jednoho VLF
< 64 MB	4	1/4 přírůstku
< 1 GB	8	1/8 přírůstku
> 1 GB	16	1/16 přírůstku

### 1.5.2 Komprese SQL databáze

Ve většině databází se nachází „řídká“ data, což znamená, že v tabulkách bývá mnoho buněk neobsazených a velké množství buněk obsahuje stejná či velmi podobná data. Jedním z řešení těchto problémů je komprese dat. Komprese dat může pomoci snížit náklady na infrastrukturu, především se to týká disků, diskových polí a úložišť. Většina dat se pravidelně zálohuje, takže efekt dosažený komprimací může být o to větší [11].

### 1.5.3 Porovnání Microsoft SQL serveru 2005 a Microsoft SQL serveru 2008

Vydání Microsoft SQL serveru 2005 bylo s nadšením vyhlášeno, protože se očekávaly velké změny. Toto vydání obsahovalo velké množství nových funkcí, včetně revizí, které zpracovávaly dotazy uvnitř databázového stroje [12].

U nové verze systému Microsoft SQL server 2008 trval vývoj mnohem kratší dobu než u verze předchozí, a proto změny v tomto systému nebyly tak drastické. Přestože celkové změny nebyly příliš rozsáhlé, tak vylepšení, kterých se nová verze dočkala, jsou důležitá a rozhodně podstatná pro organizaci [12].



Mezi nejvýznamnější novinky v Microsoft SQL serveru 2008 patří:

- optimalizovaná instalace a konfigurace,
- správa zdrojů,
- prediktivní systém optimalizace výkonu,
- rozšířená správa událostí,
- transparentní šifrování údajů,
- auditování [8].

Já se zaměřím na novinky v oblasti komprese údajů a záloh, protože jsou podstatou k dosažení cíle mé bakalářské práce. Microsoft SQL server 2008 umožňuje komprese údajů a záloh, a proto můžeme ušetřit nejen místo na disku, ale i čas [8].

## **1.6 Počítačový Server**

Spousta lidí si myslí, že nejcennější věcí v celé síti jsou počítače, avšak není tomu tak. To nejdražší, co v sítích můžeme nalézt, jsou data. Ta samozřejmě musí být někde uložena, a k tomu využíváme právě počítačový server. Počítačový server musí zaručit, aby se data na něm uložená nedostala do nesprávných rukou, proto na něj klademe vyšší nároky než na běžnou počítačovou stanici. Hlavní komponenty, které tvoří počítačový server, jsou:

- mikroprocesor,
- operační paměť,
- pevný disk [13].

### **1.6.1 Mikroprocesor**

Je označován jako mozek serveru. U levnějších strojů se používají běžné desktopové procesory např.: Intel Core i7 nebo AMD Phenom. Většinou se ale používají mikroprocesory vyvinuté speciálně pro využití v serverech jako Intel Xeon nebo AMD Opteron. Ve velkých sítích se používá multiprocessing, což označuje spolupráci několika procesorů, které jsou složeny z více jader na jedné základní desce [13].

### 1.6.2 Operační paměť

Tato komponenta serveru musí splňovat přísná kritéria. Její velikost by měla být dostatečná pro práci se soubory síťového operačního systému, pro další aplikace požadované uživateli a pro cachování obsahu pevných disků. Velikost potřebné paměti je závislá na:

- operačním systémem,
- počtu pracovních stanic,
- využívaném softwaru [13].

V dnešní době se používají operační paměti v provedení registered a ECC [13].

**Cachování:** Dříve čtená data jsou stále uchovávána v operační paměti. Pokud přijde požadavek na čtení dat, tak jsou nejprve hledána v paměti, která je rychlejší, a když zde nejsou, tak jsou čtena z pevného disku, který je pomalejší. Jelikož spousta uživatelů využívá stejný program uložený na serveru, tak pravděpodobnost nalezení dat v operační paměti je vysoká, což nám zrychlí práci serveru. Je pravda, že v dnešní době začal rozmach SSD disků, které se svou rychlostí vyrovnají paměti, avšak v serverech se většinou nenasazují, kvůli jejich vyšší ceně a nižší životnosti [13].

**ECC (Error Checking and Correcting):** Tato zkratka označuje samoopravný kód, jenž dokáže odhalit i opravit jednobitovou či dvoubitovou chybu v paměti. ECC musí být podporován základní deskou a paměťovým modulem [13].

**Registered:** Tyto moduly zvyšují stabilitu a spolehlivost přenosu dat. Registered musí být také podporovány základní deskou [13].

### 1.6.3 Pevné disky a disková pole

Disková pole označují skupinu disků, která se navenek tváří jako jeden disk. Jejich účelem je zvýšení kapacity, avšak hlavním úkolem je zvýšení bezpečnosti dat [13].

Pevné disky jsou komponentou, na níž se ukládají data z celé sítě. U levnějších serverů se používají disky připojené přes rozhraní SATA. Pokud však chceme kvalitu, tak je lepší zvolit disky s rozhraním SAS. Jedná se o konektor pro připojení pevných disků

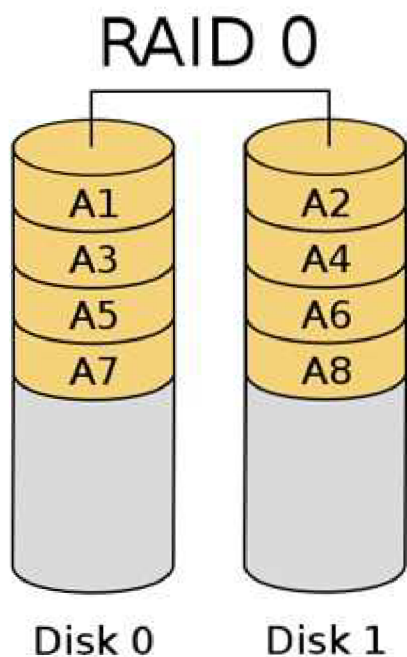
kompatibilní s rozhraním SATA, avšak je univerzálnější, rychlejší, duplexní a převážně se využívá v počítačových serverech [13].

Na disky jsou kladeny velké požadavky, co se týká bezpečnosti. Z toho důvodu se hojně využívají disková pole RAID (Redundant Array of Independent Disks). RAID je metoda, která zabezpečuje data proti selhání pevných disků, pomocí ukládání dat mezi více disků [13].

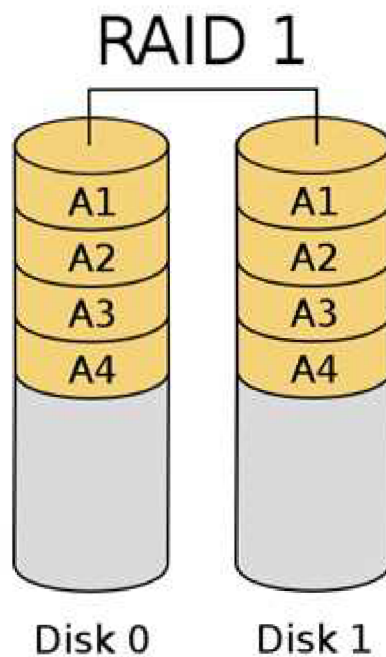
RAID se dělí do několika skupin, které jsou rozepsány v následující tabulce, a které popisují dané obrázky.

**Tabulka č. 3: Typy diskových polí RAID [13]**

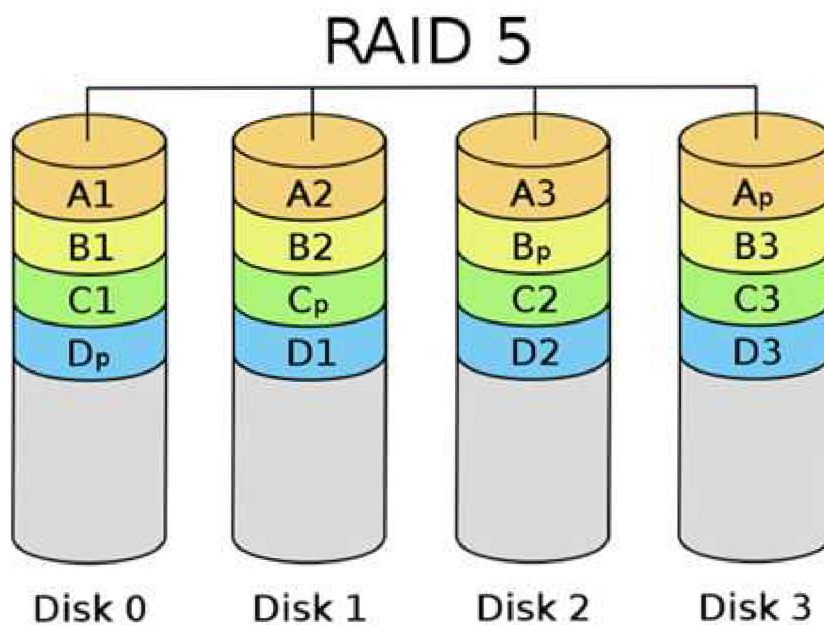
Typ	Princip	Výhody	Nevýhody
<b>RAID 0, striping</b>	Data jsou rozdělena mezi několik disků.	Zvýšení kapacity, snížení přístupové doby při čtení i zapisování.	Vůbec nezvyšuje bezpečnost, při havárii jednoho disku přicházíme o data.
<b>RAID 1, mirroring</b>	Data se současně zapisují na více disků. Jeden disk je přesnou kopií druhého.	Data jsou 100% redundantní. Dochází ke zvýšení čtecích operací díky současnému čtení ze dvou disků.	K uložení dat jsou potřeba dva disky.
<b>RAID 5, striping s redundancí</b>	Data jsou rozprostřena mezi více disků. Nadbytečná (paritní) data jsou rozdělena mezi všechny disky. Zhavarovaný disk je možné zrekonstruovat pomocí paritních dat.	Zvyšuje výkon při čtecích operacích. Redundantní data zabírají jen část kapacity disku.	Potřeba minimálně tři disků.
<b>RAID 10 striping s mirroringem (RAID 01)</b>	Data jsou rozdělena mezi několik disků (RAID 0), díky tomu se dosáhne větší rychlosti. Každý z disků RAID 0 je navíc zrcadlen (RAID 1).	Vysoká rychlost kombinovaná s bezpečností.	Na striping potřebujeme dva disky a na mirroring také, dohromady jsou tedy potřeba čtyři disky.



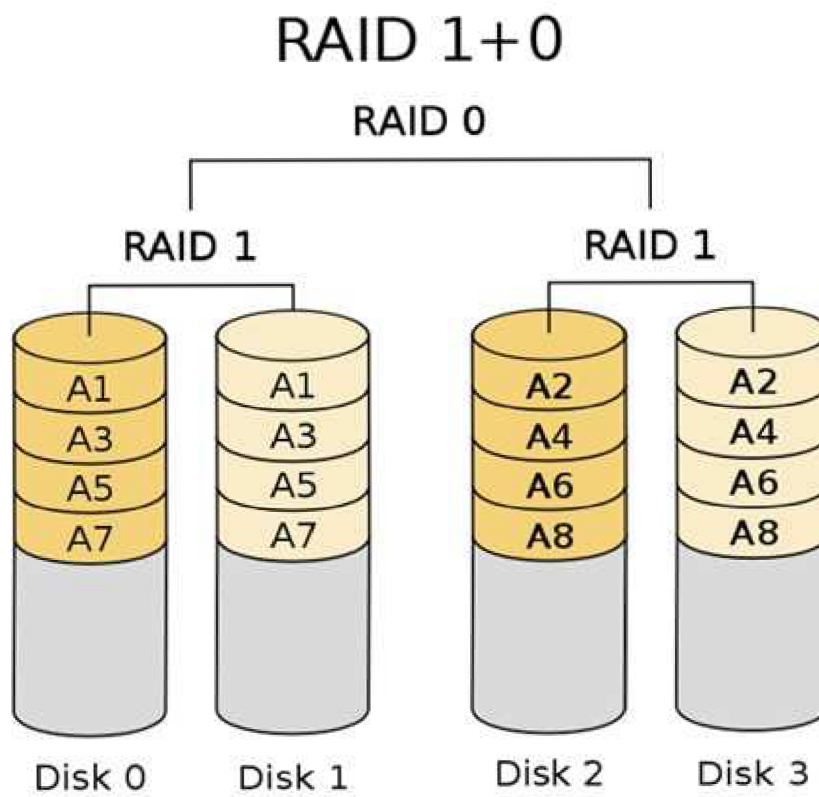
Obrázek č. 8: RAID 0, striping [14]



Obrázek č. 9: RAID 1, mirroring [14]



Obrázek č. 10: RAID 5, striping s redundancí [14]



Obrázek č. 11: RAID 10, striping s mirroringem [14]

## 2 ANALÝZA SOUČASNÉHO STAVU

V této části popíši mnou vybranou společnost, ve které jsem daný problém řešil. Dále bude detailně rozebrán stav, do kterého se databáze dostala a požadavky zákazníka na kompresi.

### 2.1 O společnosti

Vybraná společnost si nepřeje být jmenována, proto ji ve své práci budu nazývat „XYZ“. Jedná se o nadnárodní společnost, která působí kromě České republiky i v dalších 5 zemích. Podnik XYZ je poměrně velký, o čemž svědčí velikost tržeb, které se pohybovaly za rok 2012 okolo 2,5 miliard korun. Při jednodenní odstávce databáze by ztráta společnosti dosahovala řádu milionů Kč. Společnost se zabývá prodejem a distribucí určitého typu zboží.

### 2.2 Hardware

Jelikož předmětem mé práce je SQL databáze, tak je jasné, že tato databáze musí být umístěna na nějakém serveru. Společnost XYZ vlastní potřebnou infrastrukturu i kvalifikované lidské zdroje pro její obsluhu, ale přesto se rozhodla využít služeb společnosti O2, od které si pronajímá servery pro kritické aplikace, jako například běh ERP. Důvodů k outsourcingu bylo několik:

- snížení nákladů na provoz,
- neustálá odborná podpora,
- kvalitní internetové spojení
- servis.

Tento server je uložený v datovém centru v Praze, které se nazývá Nagano. Společnost si pronajímá server, který nese modelové označení HP ProLiant SL2500 Scalable System.

**Tabulka č. 4: Specifikace serveru [15]**

<b>Rodina procesorů</b>	Intel® Xeon® E5-2600 v2 product family
<b>Maximální počet procesorů</b>	12
<b>Dostupné jádro procesoru</b>	4 or 6 or 8 or 10 or 12
<b>Provedení</b>	2U
<b>Typ napájení</b>	Common slot
<b>Maximální velikost paměti</b>	512GB
<b>Paměťové sloty</b>	16 DIMM slots
<b>Typ paměti</b>	DDR3 RDIMM, LRDIMM, or UDIMM
<b>Popis jednotky úložiště</b>	(12) LFF SAS/SATA/SSD or (24) SFF SAS/SATA/SSD
<b>Síťový adaptér</b>	Embedded NIC - Intel's x2 1G NIC on board
<b>Řadič úložných zařízení</b>	Dynamic Smart Array B120i; Applicable to all models
<b>Správa infrastruktury</b>	iLO Management Engine, Insight Control

Z tabulky specifikace serveru lze vyčíst jednotlivé parametry, které server může mít. Server společnosti XYZ, na kterém běží databáze, nevyužívá maximální možné parametry.

Pronajímaný server od společnosti O2 pohání osm čtyřjádrových procesorů. Je osazen šestnácti paměti RAM, kde každá paměť má kapacitu 32GB, což dohromady dává 512GB RAM paměti. Server obsahuje také dvanáct 2,5“ pevných disků připojených přes rozhraní SAS, kde každý z disků má kapacitu 500GB. Tyto disky jsou zapojeny v RAID 10, tudíž výsledná reálná kapacita pro ukládání dat činí 3TB. V serveru jsou použity kvalitní serverové disky, které vykonají 10000 otáček za minutu.

### 2.2.1 Datové centrum Nagano

Datové centrum Nagano vlastní společnost O2 a nachází se v Praze Strašnicích. Výhody využití služeb datového centra pro outsourcing jsou uvedeny v předchozí kapitole. Kromě nich se jedná i o to, že servery jsou připojeny přímo k páteřní síti, což je vysokorychlostní síť vedoucí do Internetu. K serveru společnosti XYZ přistupují zákazníci z celého světa, proto je nutné mít kvalitní a stabilní připojení, které datové centrum Nagano nabízí. Celkově je možnost využití datového centra méně náročné nejen finančně, ale také na využití lidských zdrojů, protože o servery se musí neustále někdo starat.

Všechny systémy datacentra Nagano jsou zálohovány, ať už se to týká napájení či klimatizací. Při výpadku proudu jsou zde připraveny záložní zdroje UPS a také automatický dieselový generátor. Chlazení v datacentru zajišťují výkonné, redundantní klimatizace [16].

Mezi nejznámější zákazníky datacentra Nagano patří ministerstvo dopravy a zemědělství, v Naganu běží i stroje zajišťující chod dálničního mýtného systému, proto si myslím, že společnost XYZ si vybrala kvalitní zázemí pro své servery [16].

**Tabulka č. 5: Technické parametry datacentra Nagano [16]**

<b>Elektrické napájení</b>	2x PDU
	Dva nezávislé okruhy napájení do každého racku
	UPS a záložní diesel generátory (obojí N+1)
<b>Provozní podmínky</b>	Regulovaná teplota (23+/-2°C)
	Regulovaná relativní vlhkost (30-70%)
	Systém detekce požáru (VESDA)
	Automatický hasicí systém inertní plyn
<b>Bezpečnost</b>	Fyzické oddělení zákaznických zařízení včetně kabeláže
	Uzamykatelný prostor
	Bezpečnostní služba 24x7
	Systém průmyslové televize
	Elektronický přístupový systém - vstupní karty
<b>Technická podpora</b>	Telefonická zákaznická podpora 24x7
	Podpora při instalaci zařízení
<b>Konektivita</b>	Online přístup ke statistice využívání služby Internet Kredit přes aplikaci Moje konto
	Dohled sítě 24x7



V předchozí tabulce jsou uvedeny veškeré technické parametry, které se týkají datového centra Nagano.

## **2.3 Software**

Na serveru, kde je uložena databáze, běží operační systém Windows server 2008 Enterprise 64bit.

Jak již název napovídá, jedná se o operační systém vhodný pro servery. Verze Enterprise má stejné vlastnosti jako Standard, avšak uživatelé navíc získají čtyři licence k serveru pro provoz ve virtuálním prostředí [17].

Jelikož naše společnost zakoupila licence od firmy Microsoft na operační systém Windows server 2008, tak zároveň zakoupila licence i na databázový systém SQL server 2005 Enterprise.

S SQL serverem spolupracuje i ERP systém, který je typu all-in-one.

## **2.4 Databáze**

Jedná se o SQL databázi pro ERP systém. Jak již bylo zmíněno, tak databáze běží na fyzickém serveru. Databáze obsahuje data dvou společností holdingu. Běžně s ní prostřednictvím ERP systému pracuje zhruba 300 uživatelů a to převážně v čase od 6:00 do 23:00. Tito uživatelé přistupují k databázi současně a jsou vzájemnými konkurenty. Tato databáze je rozdělena do dvanácti různých souborů z důvodu rozložení zátěže mezi více disků a kvůli zvýšení rychlosti. Ke zvýšení zabezpečení ztráty dat se využívá na diskovém poli serveru RAID 10, a současně se asynchronně provádí zrcadlení celého prostředí z datového centra na lokální server uložený v sídle podniku XYZ. Databáze neustále běží, protože je na ni napojena kritická aplikace v podobě ERP systému.

### **2.4.1 Obsah databáze**

Databáze obsahuje naprosto vše, co je obsaženo v informačním systému společnosti. Jak již bylo řečeno, tak v databázi jsou obsažena data ze dvou společností holdingu, konkrétně České republiky a Slovenska.

V databázi jsou uvedeny veškeré informace o nákupech, prodejkách, financích, marketingu, skladových zásobách, lidských zdrojích atd.

### 2.4.2 Noční a víkendový provoz databáze

V noci, což znamená od 23:00 do 6:00, se provádí údržba dat. Jedná se o údržbu dat:

- v rámci ERP systému - jde o běžné procesní dávky, jako je přepočítání hodnot skladu a další výpočty, které se váží k provozu předešlého dne. Mezi tyto výpočty z předešlého dne lze zařadit definice slev, bonusů atd.
- v rámci databáze – sem patří částečná optimalizace indexů nebo přepočítání statistik.

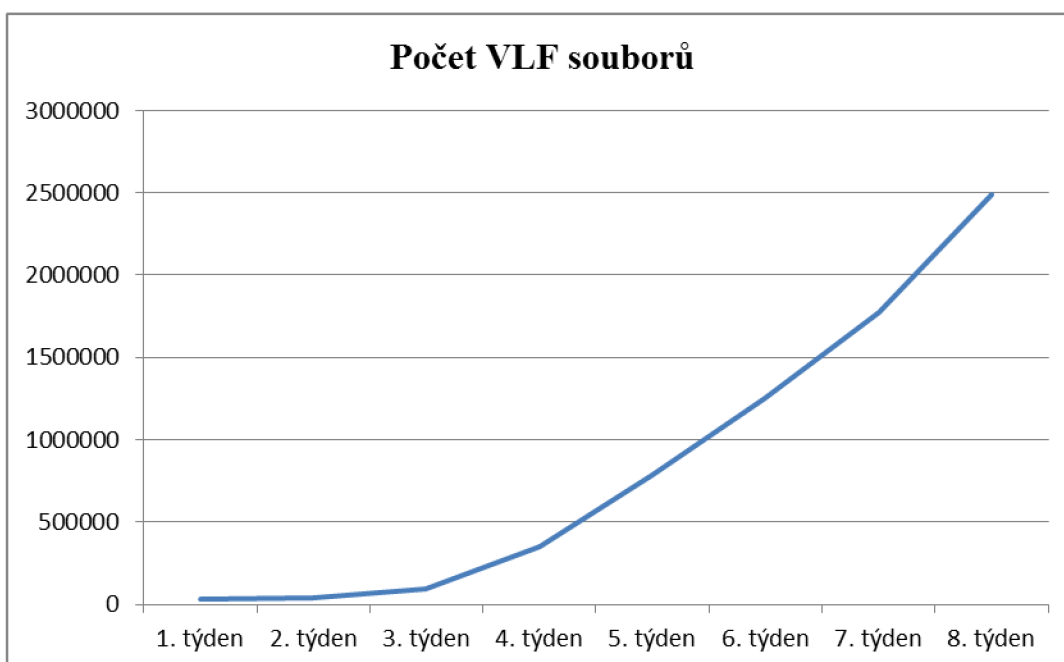
Veškeré tyto operace se neprovádí za provozu z výkonových důvodů. Zároveň se každou noc provádí záloha databáze. Tato záloha probíhá ve formě úplné zálohy celé databáze.

Další operací, která se provádí v době snížené zátěže na serveru, tudíž v noci nebo o víkendech je defragmentace databáze, díky které lze rychleji přistupovat k uloženým datům.

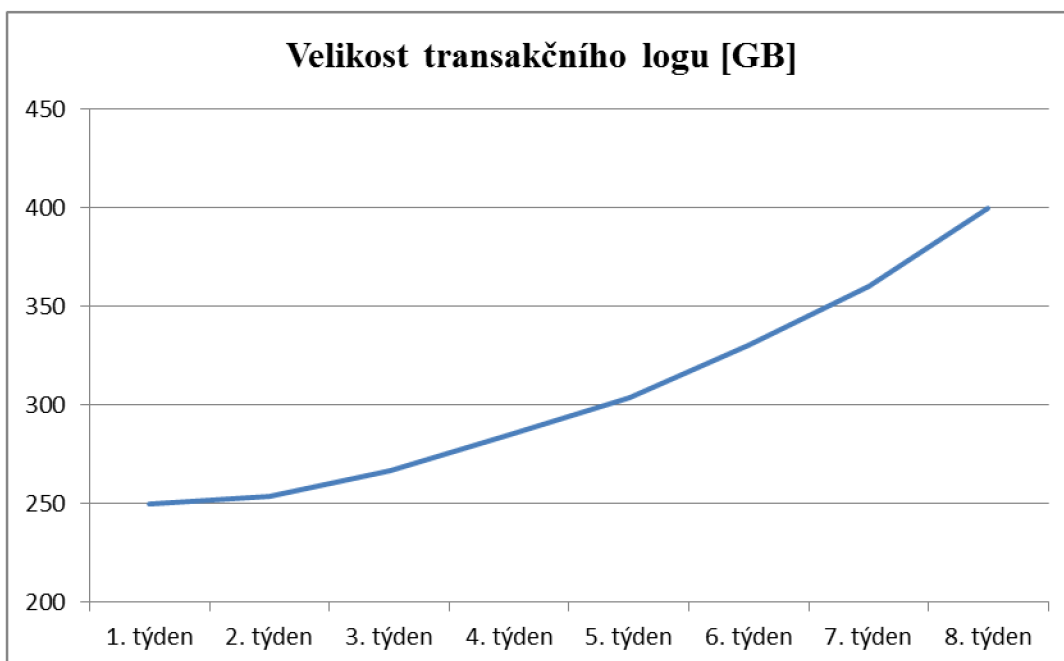
### 2.4.3 Aktuální vývoj a stav databáze

Přibližně dva měsíce zpět se databáze nacházela ve stavu, kdy její velikost oscilovala kolem hodnoty 1,85TB, z toho transakční log tvořil 250GB. Záloha databáze, která se prováděla každou noc, trvala průměrně okolo tří hodin. Obnova databáze trvala přibližně stejně dlouhou dobu jako záloha. Transakční log tvořily desítky tisíc VLF souborů.

V současné době se databáze nachází ve stavu, kdy práce s ní už téměř není možná, protože veškeré operace trvají nepřiměřeně dlouho. V této fázi dosahuje velikost databáze 2TB, z toho 1,6TB zabírají samotná data a zbylých 0,4TB vyplňuje transakční log. Doba běžných operací jako jsou záloha či obnova databáze se zvýšila. Záloha databáze se prodloužila z původních tří hodin na zhruba šest až osm hodin a obnova, která dříve trvala také tři hodiny, se prodloužila na čtrnáct hodin. Velmi také vzrostl počet VLF souborů, který se v současné době pohybuje v řádech milionů.



**Graf č. 1: Počet VLF souborů [Zdroj: Firemní dokumenty]**



**Graf č. 2: Velikost transakčního logu [Zdroj: Firemní dokumenty]**

Z předchozích grafů lze vyčíst, jak za poslední dva měsíce narostl počet VLF souborů, respektive jak narostla velikost transakčního logu. Mezi grafy je určitá podobnost a to ta, že oba rostou exponenciálně. Tato podobnost není náhodná, je to způsobeno tím, že velikost transakčního logu a počet VLF souborů na sobě závisí, protože transakční log je tvořen právě z VLF souborů.

Přesná příčina, proč databáze došla do takového stavu, není známa. Na to, že obnovení současné databáze trvá přibližně čtrnáct hodin, se přišlo díky chybě na serveru, kvůli které se databáze musela restartovat. Díky tomu se také odhalil počet VLF souborů, který se pohybuje, jak jsem již zmínil, v řádech milionů. Tento vysoký nárůst VLF souborů pravděpodobně způsobil špatně nastavený autogrow, který udává, kolik VLF souborů se má vytvořit při určitém zvětšení transakčního logu. Při tomto restartu databáze se zároveň přišlo na to, že nefunguje kompletní mirroring prostředí, jak je popsáno výše, jednoduše řečeno server, na kterém jsou všechna data zrcadlená, nepřevzal kontrolu při výpadku hlavního serveru. Po těchto zjištěních bylo nutné provést shrink transakčního logu, aby se databáze dostala do původního stavu, který byl zhruba před dvěma měsíci. Tato operace byla provedena nejprve na jiném serveru a na základě úspěšných testů se kroky zopakovali také na provozní databázi, což vedlo k tomu, že záloha a obnova databáze díky shrinku transakčního logu trvá přibližně dvě hodiny. Aby se předešlo podobnému výpadku, zákazník přijal určitá administrativní opatření a současně požaduje zmenšení velikosti databáze.

## 2.5 Požadavky zákazníka

Jak jsem již zmínil výše, tak hlavním požadavkem zákazníka bylo zmenšení velikosti databáze zhruba na 500GB. Hlavními důvody tohoto požadavku jsou:

- zrychlení veškerých operací s databází, jako je zálohování či obnovení,
- zrychlení při práci v databázi,
- celkové pročištění databáze.

Zaměstnanci zákazníka současně trvali na zachování co nejdelší historie dat. Zákazník nám kvůli vysokým ztrátám nepovolí delší odstávku databáze. Jelikož se jedná o nadnárodní společnost, tak odstávka ve státní svátky nepřipadá v úvahu, protože v každé zemi jsou jiné svátky. Odstávka o víkendech je také málo pravděpodobná, protože společnost má otevřené pobočky i o víkendech.

### 3 VLASTNÍ NÁVRHY ŘEŠENÍ

V této části své práce podám návrhy na řešení daného problému a vyberu jeden, který bude dle mého názoru pro řešení dané situace optimální.

Mnou navrhovaná řešení jsou:

- online komprese,
- jednorázový odsun dat,
- start nové databáze.

#### 3.1 Online komprese

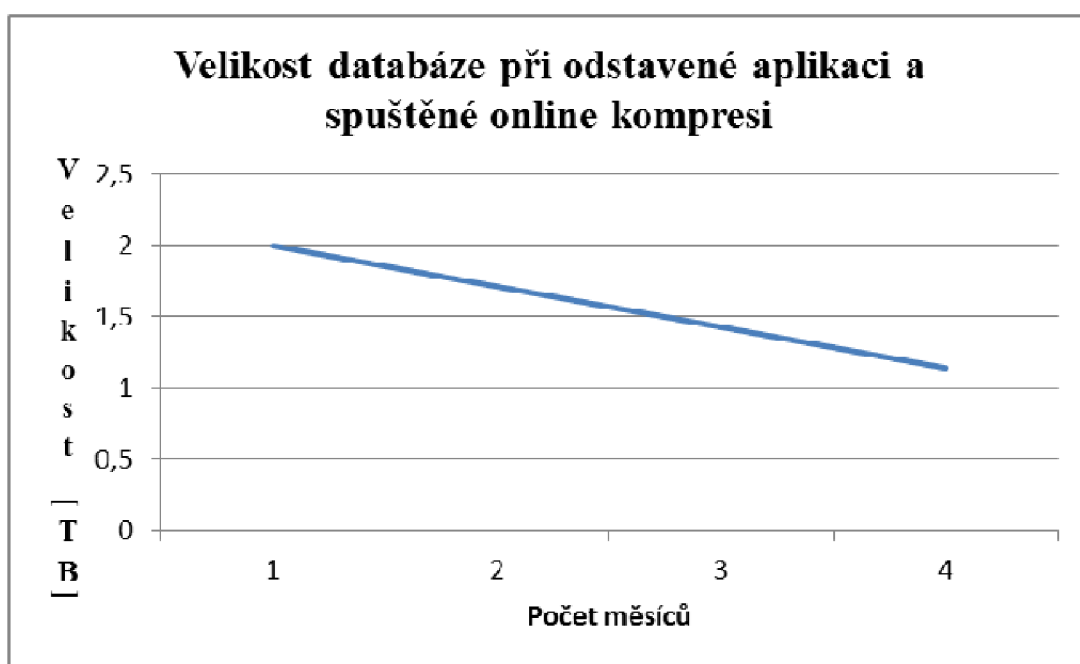
První metoda, která lze použít ke zmenšení velikosti databáze, je takzvaná online komprese.

Touto kompresí nemám na mysli novou funkci, kterou obsahuje SQL server 2008. Ale jde o online kompresi na úrovni aplikační logiky, zatímco funkce v SQL serveru nabízí kompresi na úrovni datové. Rozdíl mezi nimi nejvýstižněji vysvětlí následující příklad se zbožím:

*Máme jeden druh zboží, který jsme nakoupili desetkrát po jednom kuse. Pokud budeme uvažovat nad datovou logikou, tak bychom mohli těchto deset nákupů sloučit do jednoho, tím pádem jsme nakoupili jednou deset kusů. Avšak druhá logika navíc zkoumá zboží na úrovni aplikace, což znamená, že ještě zjišťujeme, kdo nám zboží dodal, za jakou cenu jsme ho nakoupili, na jaký sklad jsme ho přijmuli atd. Z důvodu zachování historie si nemůžeme dovolit sloučit zboží, pokud jsme ho pořídili za rozdílnou nákupní cenu.*

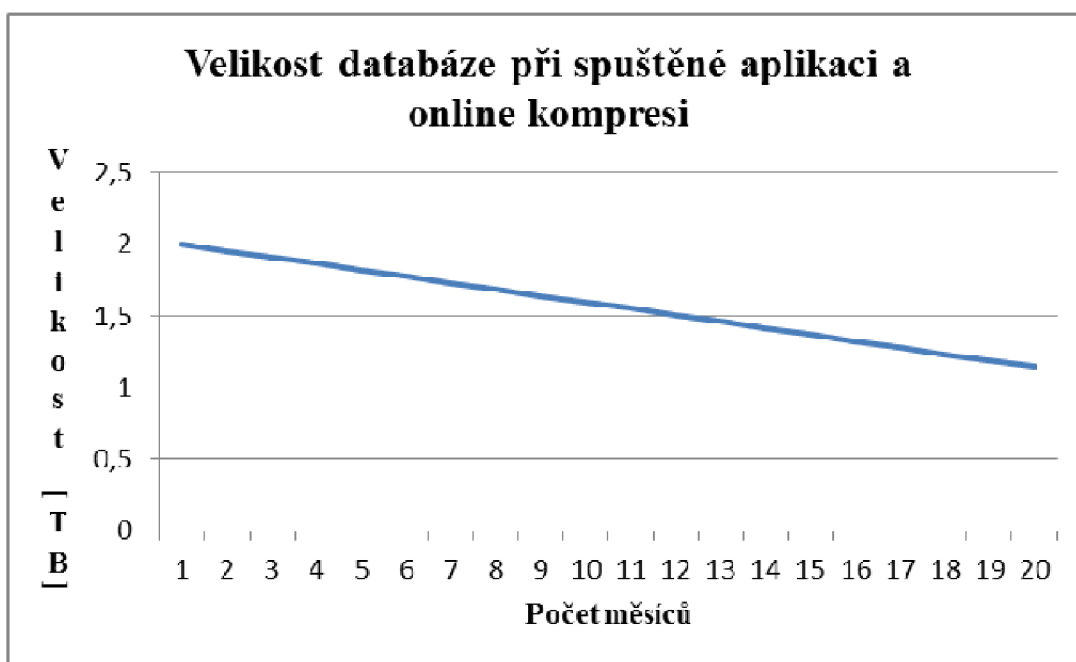
Aktualizace operačního systému Windows lze nastavit, aby se prováděly v době, kdy je na počítači nulová nebo téměř žádná aktivita. Podobný princip navrhuji použít i u online komprese, která by se prováděla pouze v době snížené zátěže na serveru, protože k jejímu provedení je zapotřebí využití mnoha systémových zdrojů. Tato doba by byla například mezi dvanáctou a jednou hodinou odpolední, kdy je snížená zátěž na serveru z důvodu menší obsluhy zákazníků, nebo mezi spuštěním běžných operací jako je zálohování nebo přepočítání hodnot skladu, které probíhají v noci.

Online komprese může probíhat téměř nepřetržitě. Za výchozí bod pro online kompresi si můžeme zvolit čas T, který označuje například datum 31. 12. 2012. Komprese se spustí nad přesně definovaným setem tabulek. Tento set určím např. dle velikosti tabulek v pořadí od největší po nejmenší. Až online komprese dosáhne času T, tak se spustí zase nad dalším setem tabulek. Tento proces se neustále opakuje do té doby, než všechny tabulky budou zkomprimované. Tato komprese by při odstavené ERP aplikaci, na základě výpočtů, trvala přibližně tři měsíce. Vzhledem k nemožnosti odstavit ERP systém na tak dlouhou dobu, z důvodu vysokých ztrát společnosti, je nutné o online kompresi uvažovat za běhu, a proto by se doba komprese dle výpočtů prodloužila na devatenáct měsíců. Po uplynutí této doby bych zase zvolil nový čas T+1, který by označoval například datum 31. 12. 2013 a opět by se opakoval stejný scénář.



**Graf č. 3: Velikost databáze při odstavené aplikaci a spuštěné online kompresi**

[Zdroj: Vlastní tvorba]



Graf č. 4: Velikost databáze při spuštěné aplikaci a online kompresi [Zdroj: Vlastní tvorba]

Předchozí grafy znázorňují rozdíl v rychlosti online komprese při odstavené aplikaci a při běžící ERP aplikaci v normální zátěži. V grafech je brán zřetel na čas, kdy může online komprese v obou případech probíhat, i na zvětšování se databáze při chodu aplikace. Online komprese při odstavené aplikaci může běžet nepřetržitě 24 hodin denně a využívat všechny systémové zdroje, což je oproti stavu, kdy by databáze byla používána, zhruba třikrát rychleji. V tomto stavu může komprese dle odhadů průměrně běžet 8 hodin čistého času denně, ale ne při využití všech systémových prostředků.

Dalším hlediskem, na které přihlédím, je neustálé zvětšování se databáze při jejím využívání. Na základě historické zkušenosti víme, že se databáze zvětšuje v průměru o 50GB za měsíc. Díky těmto faktorům doba komprese při spuštěné aplikaci naroste ze tří měsíců na devatenáct. U obou případů zkomprimujeme databázi ze 2TB na 1,145TB.

### 3.1.1 Výhody online komprese

Mezi hlavní výhody této metody patří kontinuální běh, což znamená, že online komprese se bude využívat tak dlouho, dokud je co komprimovat. Dle odhadů by se mohla současná databáze zkomprimovat při odstavené aplikaci až na požadovaných 500GB. Avšak na tuto hodnotu se pomocí online komprese při běžící aplikaci reálně



nikdy nelze dostat, protože databáze se neustále rozrůstá, což nám demonstrují předchozí grafy. Při použití této metody by nedošlo k žádné odstávce databáze, což je také velké plus, protože by při denní odstávce databáze, a tudíž i aplikace, měla společnost ztrátu v řádech milionů korun. Tato metoda nevyžaduje také zapojení tolika zdrojů na straně zákazníka. Poslední neméně důležitou výhodou je cena, která by byla nejnižší oproti ostatním řešením.

### 3.1.2 Nevýhody online komprese

Jak již uvedené grafy naznačily, tak hlavní slabinou této metody je doba, za kterou by se databáze mohla zkomprimovat na hodnoty, které se blíží požadované velikosti. Není možné provést dostatečnou očistu od nepotřebných starých dat. To je způsobeno tím, že se musí zachovat konzistence a vazby v rámci aplikační logiky v databázi. Další nevýhodou je zvýšená režie ať už lidská, tak i hardwarová. Tato komprese neustále vytěžuje veškeré zdroje serveru, tudíž server bude náchylnější na přetížení a s tím spojené poruchy hardwaru. Online kompresi je nutné často sledovat a kontrolovat, proto je potřeba vyčlenit alespoň dva členy týmu pro dohled nad průběhem komprese.

## 3.2 Jednorázový odsun dat

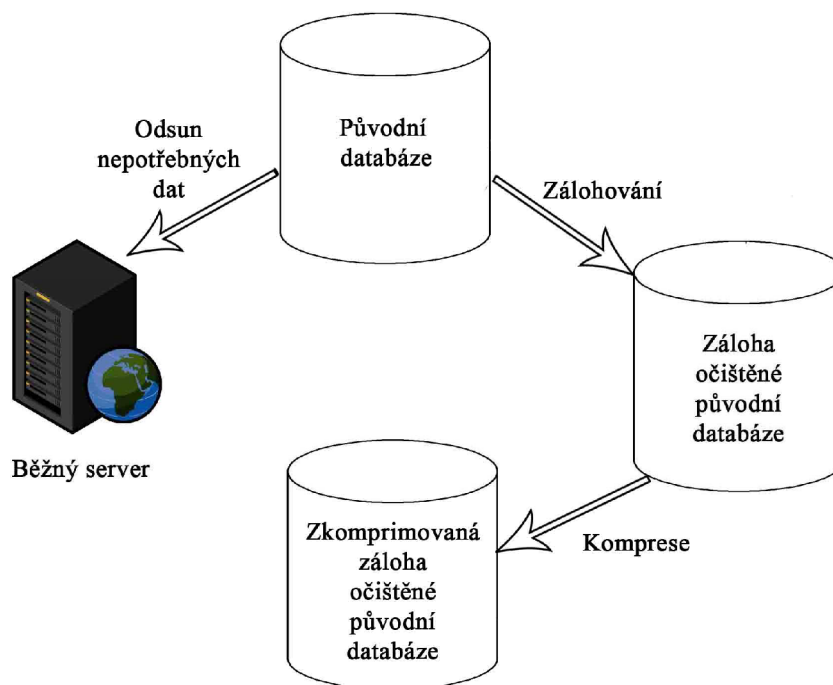
Mým druhým návrhem na kompresi databáze je jednorázový odsun dat, Tento návrh výstižně popisují následující obrázky.



Obrázek č. 12: Struktura původní databáze [Zdroj: Vlastní tvorba]

Tento obrázek popisuje, jaká data databáze obsahuje. Statická data zahrnují data o zákaznících, produktech, materiálech, dodavatelích, případně dalších obchodních entitách a definují jejich vlastnosti a parametry. Živá data vyjadřují změny a pohyby zboží či změny stavu podnikových zdrojů a kapacit. Nepotřebná data obsahují různé logy.

## Operace v čase T

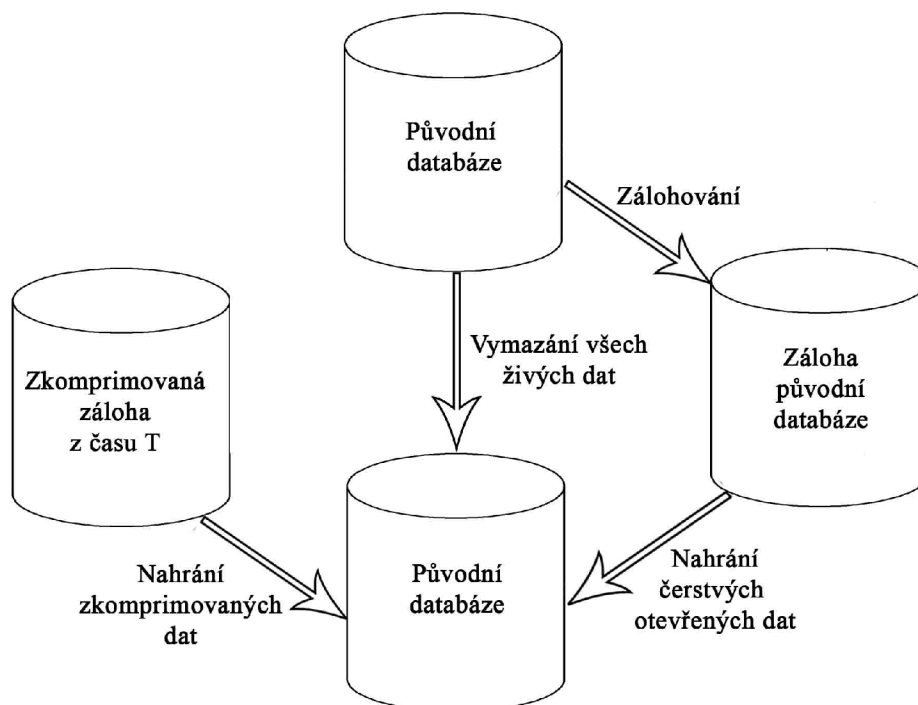


Obrázek č. 13: Operace v čase T [Zdroj: Vlastní tvorba]

Data, která jsou nepotřeba pro běžný provoz aplikace, bych odsunul na jiný server, který by měl stejné diskové pole jako náš primární. Po tomto odsunu by se data zkopírovala na nějaký běžný server, kde by byla archivována. Běžným serverem mám na mysli stroj pohybující se v cenové relaci kolem 30000Kč. Teoreticky by se tato data mohla rovnou zkopírovat na běžný server, avšak kvůli ušetření času je lepší použít stroj se stejnými parametry, jako má primární server a až poté je v klidu přesunout na běžný server.

Po odsunu nepotřebných dat vytvořím zálohu kompletní databáze. Čas, ve kterém bude tato záloha vytvořena, označím písmenem T. Veškerá data z této zálohy, která jsou starší než 31. 12. 2012, zkomprimuji pomocí online komprese, kterou jsem popsal výše. Jak lze vyčíst z grafu č. 4, tak tato komprese by odhadem trvala dva měsíce, protože by se komprimovalo menší množství dat, než je tomu v uvedeném grafu č. 4. To je způsobeno tím, že budou již nepotřebná data odstraněna.

## Operace v čase T+1



Obrázek č. 14: Operace v čase T+1 [Zdroj: Vlastní tvorba]

Po dokončení online komprese si označím datum, kdy se bude provádět převod zkomprimovaných dat to původní databáze. Toto datum ponese označení T+1. Samotná databáze by samozřejmě mezi časy T a T+1 byla v běžném provozu. Teoreticky bych nově vytvořená data, která vznikla mezi časy T a T+1, mohl také zkomprimovat, avšak kvůli zachování čerstvosti či využitelnosti některých dat je lepší je nechat v nezkomprimovaném stavu.

V čase T+1 opět udělám zálohu původní databáze. Tato záloha se uschová jako poslední záloha původní databáze pro případné budoucí dotazy či nejasnosti. Všechny tabulky týkající se živých dat vymaži a nahraji do nich zkomprimované tabulky ze zálohy z času T, což jsou tabulky týkající se dat z let 2007 až 2012, a nové nezkomprimované tabulky ze zálohy z času T+1, které se týkají dat z let 2013. Tato data se sloučí díky aplikační logice ERP systému. K vymazání a nahrání lze použít dvojice SQL příkazů DROP a CREATE, nebo DELETE a INSERT. Já bych doporučoval využít příkazů DROP a

CREATE z důvodu rychlosti. Příkaz DROP je oproti DELETE mnohem rychlejší, vymaže totiž celou tabulku, zatímco příkaz DELETE maže jednotlivé záznamy v tabulce a tuto operaci zaznamenává do transakčního logu, což u tabulek s tisíci záznamy může trvat dlouho.

Statická data	Zkomprimovaná data z let 2007 - 2012	2013
---------------	--------------------------------------	------

Obrázek č. 15: Struktura výsledné databáze [Zdroj: Vlastní tvorba]

Data, která obsahuje výsledná databáze, zobrazuje výše uvedený obrázek.

### 3.2.1 Výhody jednorázového odsunu

Jednorázový odsun je méně náročný na využití zdrojů a to jak lidských, tak i hardwarových. Oproti online kompresi by se nezvyšovala zátěž na serveru, kromě kopírování zálohy. Při tomto návrhu sice je zapotřebí využít lidské zdroje jak na naší straně, tak i na straně zákazníka, avšak počet těchto zdrojů nepřesáhne pět lidí u ani jedné ze zainteresovaných stran. Tato metoda je také mnohem rychlejší nežli online komprese. Dle výpočtů by se doba komprese vůči online kompresi zkrátila zhruba osmkrát. Jako poslední výhodu bych uvedl vyšší míru pročištění oproti první zmiňované metodě, což je způsobeno odsunem nepotřebných dat pro provoz aplikace.

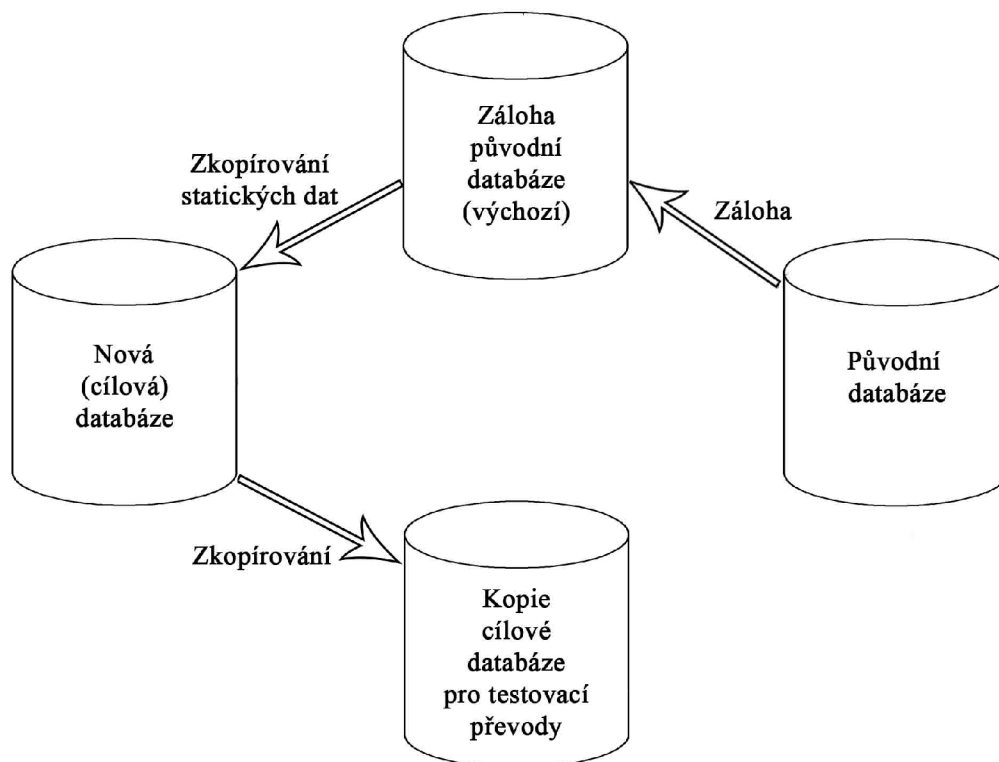
### 3.2.2 Nevýhody jednorázového odsunu

Míra očištění dat je sice vyšší než u online komprese, avšak vůči poslednímu návrhu je stále na nízké úrovni. To je způsobeno vzájemnou návazností dat na sebe, protože je nemožné smazat některá historická data, která se vážou na určitá aktuální data z důvodu konzistence databáze. Vyšší míra pročištění může být výhodou i nevýhodou z důvodu větší ztráty historických dat. Může nastat situace, kdy do společnosti XYZ zavolá zákazník a bude požadovat tu samou věc, jakou si koupil před pěti lety. Prodejce však tuto informaci již nebude znát z důvodu komprese historických dat. Na závěr bych uvedl jako nevýhodu cenu. Ta bude vyšší než u online komprese a to především proto, že samotná online komprese se využívá při kompresi dat důležitých pro provoz aplikace.

### 3.3 Start nové databáze

Posledním návrhem na řešení daného problému je start nové databáze. Start nové databáze popisují následující obrázky, které tento návrh dělí na přípravu ostrého převodu a na ostrý převod.

#### Příprava na ostrý převod

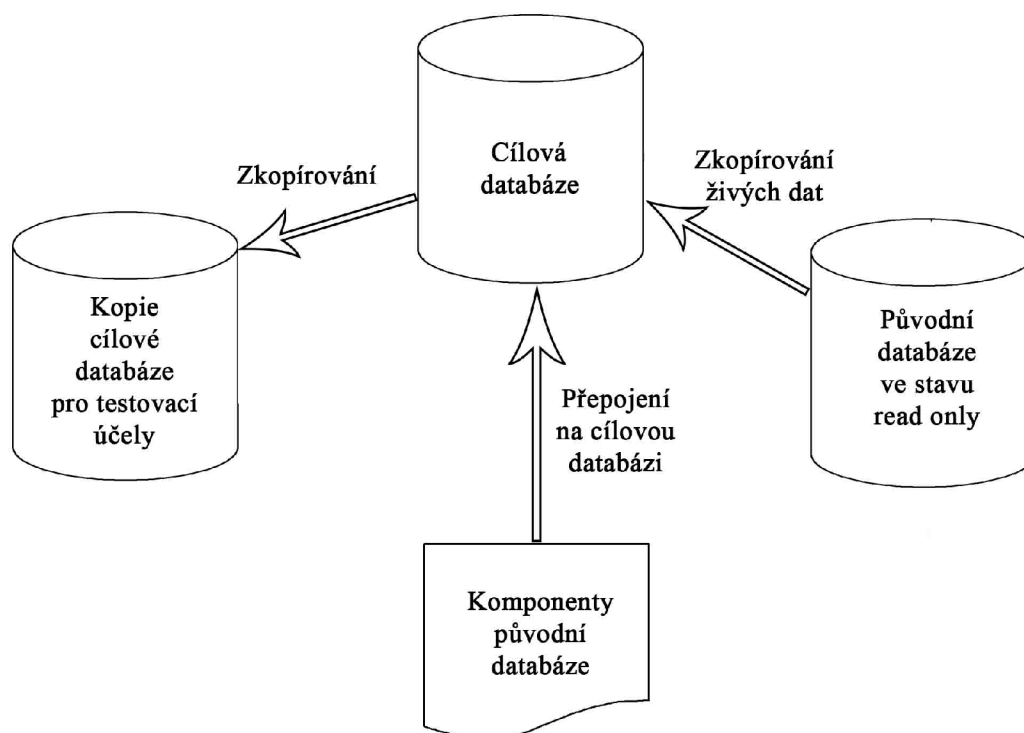


Obrázek č. 16: Příprava na ostrý převod [Zdroj: Vlastní tvorba]

Nejprve vytvoříme novou, prázdnou databázi, která na samém konci nahradí původní databázi. Struktura této databáze bude naprosto shodná s nahrazovanou databází, avšak databáze poběží na novém prostředí vytvořeném v Microsoft SQL serveru 2008. Ihned po vytvoření uděláme její kopii, se kterou budeme pracovat při testovacích převodech. Dalším krokem je otestování převodu dat do zmíněné kopie z důvodu odladění veškerých chyb, aby později tento převod proběhl hladce. Tato operace bude provedena pomocí SQL skriptů a batchů vytvořených v ERP aplikaci. Samotná operace startu nové databáze následně začne krokem zkopírování zálohy nahrazované databáze z určitého data, které označím písmenem T. Takto vytvořenou zálohu pojmenuji „výchozí

databází“ a dále budu pracovat pouze s tímto termínem. Z výchozí databáze zkopíruji statická a neměnná historická data do databáze vytvořené v prvním kroku, kterou budu nazývat „cílovou databází“. Co obsahují statická data, jsem již zmínil u předchozí metody. Z důvodu co největšího zkrácení času kopírování dat při ostrém převodu provádíme tento krok již v čase T. Tato kopírovaná data musí být pečlivě vybrána a musí jít o data taková, která nebudou uživatelé měnit. Při ostrém převodu v čase T+1 budeme kopírovat pouze otevřená živá data, protože hlavní část dat již byla odkopírována dříve. Dalším bodem bude vytvoření totožné kopie cílové databáze, kterou budeme používat pro testovací převod otevřených živých dat a následných uživatelských testů. Na této testovací databázi se vyzkouší veškeré přesuny z výchozí databáze. Těmito přesuny mám na mysli kompresi či kopírování. Tento krok se provádí opět z důvodu hladkého průběhu při ostrém převodu. Jakmile bude cílová - testovací databáze naplněna živými daty z výchozí databáze, bude tato databáze předána koncovým uživatelům pro testy a ověření. V tomto kroku se odhalí, zda-li vše bylo navrženo a provedeno správně, případně se upraví převody dle nových požadavků od uživatelů. Původní - nahrazovaná databáze samozřejmě při přípravách normálně dále běží.

## Ostrý převod



Obrázek č. 17: Ostrý převod [Zdroj: Vlastní tvorba]

Při započetí ostrého převodu, v čase  $T+1$ , dojde k zastavení činnosti v původní – nahrazované databázi. To zajistíme jejím přepnutím do stavu read only. Následným krokem bude zkopírování otevřených živých dat z nahrazované read-only databáze do vytvořené cílové databáze, naplněné statickými daty z času  $T$ . Do cílové databáze budeme tedy kopírovat pouze otevřená živá data. Pojem otevřená živá data nejlépe vystihuje následující příklad:

*V ERP systému máme fakturu, kterou nám doposud zákazník nezaplatil. My však do cílové databáze nebudeme převádět celou fakturu, nýbrž jen částku, kterou nám zákazník dluží, díky čemuž ušetříme v databázi místo.*

Tento převod by byl proveden o víkendu z důvodu co nejmenší ztráty společnosti. Ostrý převod má několik kroků:

- Samotný převod živých dat, u kterého, jak jsem již zmínil, se zkopírují pouze otevřená živá data.
- Přepojení všech komponent v prostředí z nahrazované na cílovou databázi, jedná se o různé externí aplikace, datové sklady, webové aplikace, aplikační servery, apod.
- Kopie celého prostředí, která se vytvoří z cílové databáze.
- Otestování kopie prostředí koncovými uživateli
- Spuštění automatických dávek na ostrém prostředí, kam patří například noční a denní automaty, jako doučtování, exporty do externích aplikací, přepočty plánu atd.
- Test ostrého prostředí

### **3.3.1 Výhody startu nové databáze**

Hlavní výhodou je vysoká míra očištění, díky které by se dle odhadů mohla původní databáze zkomprimovat na velikost 500GB. Při této metodě můžeme původní databázi očistit od nepotřebných dat maximálním způsobem. Toho je docíleno díky tvorbě nové databáze, kam by se nahrála pouze nezbytná historická data a ostatní by se nastavila jako počáteční stavy. V databázích vlivem dlouhého používání mohou vzniknout určité nekonzistentní vztahy. Ty by při této metodě byly odstraněny právě díky tvorbě úplně nové databáze.

### **3.3.2 Nevýhody startu nové databáze**

Velká nevýhoda, která by odradila mnoho zákazníků od volby této metody, je cena. Je pochopitelné, že cena bude vyšší. Je to způsobeno tím, že se vytváří kompletně nová databáze a nové prostředí, na kterém poběží. Dalším důvodem zvýšení ceny je detailní analýza a návrh řešení, jak dostat data ze staré databáze do nové. Při aplikaci tohoto řešení je zapotřebí mnoho lidských zdrojů jak na straně zákazníka, tak na straně vykonavatele. Na straně zákazníka je to způsobeno především testováním, které je však kvůli hladkému průběhu nezbytné. Na straně vykonavatele to zase způsobuje detailní



analýza a samotná realizace. Ztráta historických dat samozřejmě může být i nevýhodou, to vysvětluje příklad, který jsem uvedl u nevýhod jednorázového odsunu. Veškeré komponenty musí být předělány, aby pracovaly se správnou databází. Hrozí zde riziko, že se na něco zapomene, a proto daná část nepojede, nebo nebude správně fungovat.

### 3.4 Výběr optimálního řešení

Optimální řešení budu vybírat ze tří návrhů, které jsem popsal výše. Nejjednodušší bude si porovnat výhody a nevýhody jednotlivých návrhů na řešení problému. Tyto atributy všech metod komprese přehledně ukáže následující tabulka.

**Tabulka č. 6: Výhody a nevýhody jednotlivých řešení [Zdroj: Vlastní tvorba]**

Řešení	Výhody	Nevýhody
Online komprese	Kontinuální běh	Doba komprese
	Žádná odstávka databáze	Nízká míra pročištění
	Nízké vytížení lidských zdrojů na straně zákazníka	Vysoké vytížení hardwarových zdrojů
	Cena	Vysoké vytížení lidských zdrojů na naší straně
Jednorázový odsun dat	Nízké vytížení lidských zdrojů na naší straně i na straně zákazníka	Vyšší míra pročištění
	Nízké vytížení hardwarových zdrojů	
	Doba komprese	Cena
	Vyšší míra pročištění	
Start nové databáze	Extremně vysoká míra pročištění	Cena
	Odstranění nekonzistentních vztahů	Vysoké vytížení lidských zdrojů na naší straně i na straně zákazníka
	Doba komprese	Ztráta historických dat

Z výše uvedených metod bych vybral jednorázový odsun dat, který bych i doporučil zákazníkovi. Vede mě k tomu především krátký čas realizace, ať už se jedná o přípravu či samotnou kompresi, a cena. Oba tyto atributy se nachází někde uprostřed mezi online kompresí a mezi startem nové databáze. Samozřejmě, že míra očištění nebude taková,

jako u startu nové databáze, ale dle odhadů by se mohla databáze zkomprimovat až na hodnotu 700GB.

Rozhodující slovo má vždy zákazník, pro kterého se tento úkol řeší. Mezi jeho požadavky patří co největší zmenšení databáze, aby se s ní zrychlila práce, a také aby veškeré pravidelné operace netrvaly tak dlouho. Primárním požadavkem však je zmenšení velikosti databáze pod 500GB. Pro splnění tohoto kritéria jsem se nakonec rozhodl vybrat metodu startu nové databáze. Tato metoda nejlépe vyřeší požadavky zákazníka, protože s ní se databáze nejvíce očistí od nepotřebných dat a díky tomu je reálné dosáhnout velikosti 500GB.

### 3.5 Aplikace vybraného řešení

Jak jsem již zmínil výše, tak vybraným návrhem na řešení, které bude aplikované na tento problém je start nové databáze. Následující diagram ukazuje nutnou časovou následnost některých úloh. Některé je možné provádět paralelně a jsou na sobě nezávislé. Délka jednotlivých úloh nijak nekoresponduje s grafickým zobrazením. Ta bude upřesněna v tabulce č. 8.

#### Časová následnost úloh při převodu databáze

K = konec práce uživatelů ve staré databázi

Z = začátek práce uživatelů v nové databázi

Proces	Provedení		K	Z
Tvorba nové databáze	Realizátor	■		
Komprese	Realizátor		■	
Převod statických, neměnných dat	Realizátor			■
Převod živých dat	Realizátor		■	
Ruční doladění převodu	Realizátor			■
Testování	Realizátor + Zákazník	■	■	
Kontrola	Realizátor + Zákazník	■	■	
Opravy	Realizátor			■
Spuštění nové databáze	Realizátor			■
Podpora po spuštění nové databáze	Realizátor			■

Obrázek č. 18: Časová následnost úloh při převodu databáze [Zdroj: Vlastní tvorba]

Rád bych objasnil jednotlivé procesy, které jsou uvedeny v předchozím obrázku.

Stará databáze běží na platformě Microsoft SQL server 2005. Novou databázi by bylo dobré vytvořit na novější platformě, konkrétně na Microsoft SQL serveru 2008 nebo 2012. Všechny tyto verze jsou mezi sebou kompatibilní, tudíž s převodem dat ze staré verze do nové by neměly nastat žádné komplikace. V nově vytvořené databázi lze využít nové funkce Microsoft SQL serveru 2008 či 2012, jako je online komprese, díky které se může velikost databáze udržovat v rozumném stavu.

Kompresi dat se bude provádět pomocí připravených SQL skriptů a pomocí aplikační logiky ERP systému. Doporučil bych rozdělit databázi na tři menší, konkrétně na českou, slovenskou a společných dat. Tímto rozdělením bychom docílili rychlejší práce s jednotlivými databázemi. Součet velikostí těchto databází bude po kompresi dle odhadů oscilovat okolo hodnoty 500GB.

Samotné převody bychom si nejprve otestovali, aby při ostrém převodu vše probíhalo hladce, protože si nemůžeme dovolit prodloužení doby převodu, kvůli ztrátám společnosti, které by v případě zdržení nastaly. Převod jak statických, tak živých dat by se prováděl také pomocí předpřipravených SQL skriptů.

Některé operace probíhající v ERP aplikaci, jako jsou nákupní či prodejní objednávky, musí být před ostrým i testovacím převodem uzavřené z důvodu konzistence dat v nové databázi. O toto uzavření, a následnou kontrolu, se postará zákazník.

Pod pojmem ruční doladění převodu se skrývají drobné operace jako přepojení externích aplikací, datových skladů, webových aplikací, aplikačních serverů, pohledů, automatů atd. na novou databázi. Po odstavení databáze je nutné opravit kompletní mirroring prostředí u zvoleného RAID 10. Mirroring na diskové pole uložené na jiném serveru bych ponechal, protože oproti mirroringu, který by se prováděl mezi diskovými poli na jednom serveru, nese řadu výhod. Největší výhodou je možnost přepojení na druhý server a to nejen při poruše hardwaru, ale i při aplikačním problému, což mirroring na jednom serveru neumožňuje. Samozřejmě toto řešení s sebou nese i nevýhody jako jsou velké množství převáděných dat, které jsou náchylné na chyby při spojení, a navýšení transakčního logu. Avšak tyto nevýhody jsou oproti výhodě

zanedbatelné. Další problém, který je nutné opravit je špatně nastavený autogrow. Ten bych nastavil dle tabulky č. 2, která je uvedena v teoretické části.

Z diagramu lze vyčíst, že testování a kontrola probíhají paralelně s přechovými operacemi. To vše se provádí z důvodu předejití nechtěným komplikacím, které by mohly při ostrém převodu nastat. Pokud se při testování odhalí nějaké nedostatky, tak se ihned opraví.

S ostrým startem souvisí podpora zákazníka. Předpokládáme, že se nepodaří při všech testech odladit kompletně všechny problémy. Z toho důvodu by měl realizátor zákazníkovi poskytnout zvýšenou podporu a s ní spojenou záruku na opravu nalezených chyb.

**Tabulka č. 7: Délka trvání jednotlivých úkonů** [Zdroj: Vlastní tvorba]

<b>Proces</b>	<b>Hodiny</b>
<b>Analýza</b>	300
<b>Příprava rozdělení databáze</b>	80
<b>Test na rozdělení databáze</b>	20
<b>Příprava skriptů na kompresi a převod</b>	250
<b>Testovací převod</b>	400
<b>Podpora při testování na straně zákazníka</b>	50
<b>Rozdělení databáze</b>	20
<b>Ostrý převod</b>	235
<b>Podpora po ostrém převodu</b>	215
<b>SUMA</b>	1570

Doba trvání jednotlivých úkonů uvedená v tabulce je pouze informativní. Není možné předpovídat veškerá rizika, která mohou při celém převodu nastat, proto se doba převodu 1570 hodin může změnit na základě detailní analýzy.

## ZÁVĚR

Cílem mé bakalářské práce je návrh na kompresi SQL databáze vztahující se k ERP systému. Jedná se o reálný problém, který vznikl v jedné nejmenované společnosti zabývající se prodejem a distribucí zboží.

Obsahem práce jsou teoretické poznatky, které jsou dále převedeny do praxe. Druhou kapitolu tvoří analýza současného stavu, kde jsou uvedeny základní údaje o společnosti, o použitém hardwaru a softwaru, ale především o databázi a problému, který nastal. Na závěr této kapitoly jsou shrnuty požadavky zákazníka, na jejichž základě jsem se rozhodl pro výběr jednoho konkrétního návrhu řešení dané problematiky.

Po těchto dvou kapitolách následuje kapitola nazvaná vlastní návrhy řešení. Obsahem této kapitoly jsou tři konkrétní návrhy na řešení problému s databází.

Prvním návrhem je online komprese, která by se prováděla za běhu databáze při malé zátěži na serveru. Hlavní nevýhodou tohoto řešení je doba trvání, která by se pohybovala v řádu let, avšak na druhou stranu by nebyla potřeba žádná odstávka databáze a tudíž by společností nevznikla žádná ztráta.

Druhým návrhem na řešení je jednorázový odsun dat. Součástí tohoto řešení je odsun databáze na jiný server, kde by se zkomprimovala a poté by zkomprimované tabulky nahradily stávající. Při této metodě by se nevyužívalo tolik hardwarových ani lidských zdrojů, na druhou stranu míra očištění dat je menší než u startu nové databáze.

Poslední je start nové databáze, kde by se zkopírovala stávající databáze, zkomprimovala se a poté by se zkopírovala do nově vytvořené databáze. Tato metoda nabízí možnost největší komprese oproti ostatním metodám, na druhou stranu za kvalitu se platí, proto je tato metoda nejdražší.

Mnou zvolená metoda je start nové databáze, protože nejvíce vyhovuje požadavkům zákazníka. Na závěr kapitoly je popsána aplikace vybraného řešení včetně návaznosti a doby trvání jednotlivých procesů.

Výsledkem mé práce bude dle odhadů původní databáze o velikosti 2000GB zkomprimovaná na přijatelných 500GB, což splňuje požadavky zákazníka, který trval na zmenšení velikosti na právě tuto hodnotu.

## SEZNAM POUŽITÉ LITERATURY

- [1] SODOMKA, Petr a Hana KLČOVÁ. *Informační systémy v podnikové praxi*. 2. aktualizované a rozšířené vydání. Brno: Computer Press, 2010, 501 s. ISBN 978-80-251-2878-7.
- [2] KOCH, Miloš, Jan DOVRTĚL a kol. *Management informačních systémů*. 3. přepracované vydání. Brno: Akademické nakladatelství CERM, 2010, 171 s. Učební texty vysokých škol. ISBN 978-80-214-4157-6.
- [3] SODOMKA, Petr a Hana KLČOVÁ. CVIS. *CVIS* [online]. 2012 [cit. 2014-04-30]. Dostupné z: <http://www.cvis.cz/hlavni.php?stranka=novinky/clanek.php&id=660>
- [4] KOREJS, Martin a Jiří RÁKOSNÍK. ERP - Dnes výhoda, zítra nezbytnost. *CIO Business World.cz | IT strategie pro manažery* [online]. 2008 [cit. 2014-04-30]. Dostupné z: <http://businessworld.cz/erp-bi-bpm/erp-dnes-vyhoda-zitra-nezbytnost-1978>
- [5] ERP systém. *Informační systémy | KEIL* [online]. 2013 [cit. 2014-04-30]. Dostupné z: <http://www.vaclavkeil.cz/erp-system/>
- [6] SKŘIVAN, Jaromír. Databáze a jazyk SQL. *Interval.cz* [online]. 2000 [cit. 2014-04-30]. Dostupné z: <http://interval.cz/clanky/databaze-a-jazyk-sql/>
- [7] CONOLLY, Thomas, Carolyn E BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.
- [8] LACKO, Ľuboslav. *Jak vyžrát na Microsoft SQL Server 2008: správa, konfigurace, programování*. Brno: Computer Press, 2009. ISBN 978-80-251-2101-6.
- [9] JECHA, Tomáš. Fyzické soubory databáze. *Největší český web zaměřený na .NET framework* [online]. 2009 [cit. 2014-04-30]. Dostupné z: <http://www.dotnetportal.cz/clanek/143/Fyzicke-soubory-databaze-datove-soubory-a-transakcni-log>
- [10] CHMEL, Marek. Princip práce SQL Server Recovery Models. *Microsoft TechNet: Materiály pro IT odborníky* [online]. 2011 [cit. 2014-04-30]. Dostupné z: <http://blogs.technet.com/b/technetczsk/archive/2011/11/01/princip-prace-sql-server-recovery-models.aspx>
- [11] LACKO, Ľuboslav. *Business Intelligence v SQL Serveru 2008: reportovací, analytické a další datové služby*. Brno: Computer Press, 2009. ISBN 978-80-251-2887-9.

- [12] WALTERS, Robert, Michael COLES a kol. *Mistrovství v Microsoft SQL Server 2008: [kompletní průvodce databázového experta]*. Brno: Computer Press, 2009. ISBN 978-80-251-2329-4.
- [13] HORÁK, Jaroslav a Milan KERŠLÁGER. *Počítačové sítě pro začínající správce*. 5. aktualizované vydání. Brno: Computer Press, 2011. ISBN 978-80-251-3176-3.
- [14] TUHÝ, Roman. NAS: Práce s daty a sdílení pro pokročilé. *Svět hardware* [online]. 2013 [cit. 2014-04-30]. Dostupné z: <http://www.svethardware.cz/nas-prace-s-daty-a-sdileni-pro-pokrocile/37490-2>
- [15] HP ProLiant Scalable Systems. *HP®* [online]. 2014 [cit. 2014-04-30]. Dostupné z: <http://www8.hp.com/cz/cs/products/proliant-servers/product-detail.html?oid=6667379#!tab=specs>
- [16] Služby datových center (Housing). *O2* [online]. 2014 [cit. 2014-04-30]. Dostupné z: [http://www.o2.cz/pa/191733-managed\\_hosting/sluzby-datovych-center.html](http://www.o2.cz/pa/191733-managed_hosting/sluzby-datovych-center.html)
- [17] Produktové verze Windows Server 2008. *SystemOnLine RSS* [online]. 2007 [cit. 2014-04-30]. Dostupné z: <http://www.systemonline.cz/sprava-it/produktove-verze-windows-server-2008-z.htm>

## **SEZNAM POUŽITÝCH ZKRATEK**

BI - Business Intelligence  
CRM - Customer Relationship Management  
DBMS - Database Management System  
DSS - Decision Support Systems  
ECC - Error Checking and Correcting  
EDI - Electronic Data Interchange  
EIS - Executive Information Systems  
E-R diagram - Entity–Relationship diagram  
ERP - Enterprise Resource Planning  
GB - Gigabyte  
IS – Informační systém  
MIS - Management Information Systems  
RAID - Redundant Array of Independent Disks  
RAM - Random-Access Memory  
SaaS - Software as a Service  
SAS - Serial Attached SCSI  
SATA - Serial ATA  
SCM - Supply Chain Management  
SQL - Structured Query Language  
TB - Terabyte  
TPS - Transaction Processing Systems  
UPS - Uninterruptible Power Supply  
VLF - Virtual Log File



## SEZNAM OBRÁZKŮ

Obrázek č. 1: IS z pohledu architektur .....	12
Obrázek č. 2: IS z pohledu úrovně řízení.....	14
Obrázek č. 3: IS z pohledu okolí.....	15
Obrázek č. 4: Holisticko-procesní pohled na podnikové informační systémy .....	16
Obrázek č. 5: Struktura ERP systému.....	17
Obrázek č. 6: SaaS .....	18
Obrázek č. 7: Systém přístupu k databázi.....	21
Obrázek č. 8: RAID 0, striping.....	28
Obrázek č. 9: RAID 1, mirroring.....	28
Obrázek č. 10: RAID 5, striping s redundancí.....	28
Obrázek č. 11: RAID 10, striping s mirroringem .....	29
Obrázek č. 12: Struktura původní databáze .....	41
Obrázek č. 13: Operace v čase T .....	42
Obrázek č. 14: Operace v čase T+1 .....	43
Obrázek č. 15: Struktura výsledné databáze .....	44
Obrázek č. 16: Příprava na ostrý převod.....	45
Obrázek č. 17: Ostrý převod .....	47
Obrázek č. 18: Časová následnost úloh při převodu databáze.....	50

## SEZNAM GRAFŮ

Graf č. 1: Počet VLF souborů .....	35
Graf č. 2: Velikost transakčního logu .....	35
Graf č. 3: Velikost databáze při odstavené aplikaci a spuštěné online kompresi .....	39
Graf č. 4: Velikost databáze při spuštěné aplikaci a online kompresi .....	40

## SEZNAM TABULEK

Tabulka č. 1: Klasifikace ERP systémů.....	19
Tabulka č. 2: Počet a velikost VLFs v závislosti na zvětšení transakčního logu .....	23
Tabulka č. 3: Typy diskových polí RAID.....	26
Tabulka č. 4: Specifikace serveru .....	31
Tabulka č. 5: Technické parametry datacentra Nagano.....	31
Tabulka č. 6: Výhody a nevýhody jednotlivých řešení .....	49
Tabulka č. 7: Délka trvání jednotlivých úkonů.....	51