

Optimalizace výkonu SAP instancí typu ABAP

Bc. Petr VÍŠEK

Diplomová práce
2015



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Petr Víšek**
Osobní číslo: **A13510**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Forma studia: **kombinovaná**

Téma práce: **Optimalizace výkonu SAP instancí typu ABAP**
Téma anglicky: **SAP ABAP Performance Optimization**

Zásady pro vypracování:

1. Popište jednotlivé SAP instance typu ABAP a clusterové řešení a jednotlivé možnosti zvyšování výkonu SAP instancí typu ABAP.
2. Definujte klíčové ukazatele výkonnosti (KPI).
3. Popište databázi SAP HANA a porovnejte s ostatními databázemi (Oracle, DB2, MSSQL).
4. Diskutujte zvýšení výkonu při přechodu na databázi SAP HANA.
5. Testujte vybrané metody zvýšení výkonu na reálném SAP systému.
6. Zhodnoťte dosažené výsledky.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. SCHNEIDER, Thomas. **SAP performance optimization guide: analyzing and tuning SAP systems**. 7th edition. 837 pages. ISBN 978-1-59229-874-7.
2. SCHNEIDER, Thomas. **SAP R/3 performance optimization: the official SAP guide**. San Francisco: Sybex, c1999, xxix, 545 s. ISBN 07-821-2563-8.
3. BREMER, Richard a Lars BREDDemann. **SAP HANA administration**. 1st edition. 722 pages. ISBN 978-159-2299-546
4. FOSE, Frank, Sigrid HAGEMANN a Liane WILL. **SAP NetWeaver AS ABAP system administration**. 4th ed. Boston: Galileo Press, 2012, 747 p. ISBN 15-922-9411-1.
5. KELLER, Horst. **The official ABAP reference**. 3rd ed. Boston: Galilo Press, c2011, p. cm. ISBN 15-922-9376-X.
6. **SAP Help Portal** [online]. Dostupné z <http://help.sap.com>
7. **SAP Community Network** [online]. Dostupné z <http://scn.sap.com>

Vedoucí diplomové práce:

RNDr. Ing. Miloš Krčmář

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

6. února 2015

Termín odevzdání diplomové práce:

15. května 2015

Ve Zlíně dne 6. února 2015



doc. Mgr. Milan Adámek, Ph.D.

děkan



L.S.



doc. Mgr. Roman Jašek, Ph.D.

ředitel ústavu

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně 14. 5. 2015


.....
podpis diplomanta

ABSTRAKT

Cílem diplomové práce je představit SAP instance typu ABAP, popsat jednotlivé možnosti zvyšování jejich výkonu a celkový proces optimalizace. Dále jsou rozebrány klíčové ukazatele výkonnosti, funkce clusteru, různé druhy pamětí a prostředky pro analýzu. Závěr teoretické části práce pojednává o databázi SAP HANA. Praktická část obsahuje konkrétní ukázky postupu při analýze výkonnostních problémů a jejich řešení.

Klíčová slova: SAP, optimalizace výkonu, analýza, SAP HANA

ABSTRACT

The aim of this diploma thesis is introduce ABAP SAP instances, describe possibilities of performance increasing and whole optimization process. Further there are described key performance indicators, cluster function, different kinds of memory and finally tools for analysis. The end of theoretical part discuss about SAP HANA database. Practical part contains concrete examples of performance analysis and it's solution.

Keywords: SAP, performance optimization, analysis, SAP HANA

Chtěl bych poděkovat vedoucímu diplomové práce, panu RNDr. Ing. Miloši Krčmářovi za veškerou pomoc, rady a podporu při zpracovávání zadaného tématu. Dále bych chtěl poděkovat mé manželce Mgr. Janě Víškové za veškerou podporu nejen při psaní této práce, ale i během celého studia a v osobním životě.

OBSAH

ABSTRAKT	5
ABSTRACT	5
ÚVOD	9
I TEORETICKÁ ČÁST	10
1 SYSTÉM SAP A TYPY INSTANCÍ	11
1.1 VYMEZENÍ POJMŮ.....	11
1.2 VRSTVY SYSTÉMU SAP.....	12
1.3 INSTANCE SAP.....	12
1.3.1 Centrální instance.....	13
1.3.2 Dialogová instance.....	13
1.3.3 ASCS instance.....	13
1.3.4 ERS instance.....	13
1.4 VÝBĚR SAP INSTANCÍ.....	13
1.4.1 Jednoduché řešení.....	14
1.4.2 Více výkonu.....	14
1.5 ZVÝŠENÍ VÝKONU INSTANCE.....	14
1.5.1 Dialog proces.....	15
1.5.2 Background proces.....	15
1.5.3 Spool proces.....	16
1.5.4 Update proces.....	16
1.6 OPERAČNÍ MÓDY.....	16
2 CLUSTER	17
2.1 ZÁKLADNÍ FUNKCE.....	17
2.2 SYNCHRONIZACE SOUBOROVÉHO SYSTÉMU.....	17
2.3 VYUŽITÍ NEAKTIVNÍCH SERVERŮ.....	18
2.4 CLUSTER BEZ INVESTIC DO HARDWARU.....	18
2.5 SÍŤOVÁ KONFIGURACE CLUSTERU.....	19
3 PAMĚŤ SYSTÉMU SAP	20
3.1 PAMĚŤOVÉ PARAMETRY.....	21
4 OPTIMALIZACE	24
4.1 TECHNICKÁ OPTIMALIZACE.....	24
4.2 APLIKAČNÍ OPTIMALIZACE.....	25
4.3 JAK ČASTO OPTIMALIZOVAT.....	25
4.4 ZDROJ PROBLÉMŮ.....	25
5 ANALÝZA ZÁTĚŽE	27
5.1 MONITOR ZÁTĚŽE.....	27

5.1.1	Průměrná doba odezvy.....	29
5.1.2	Doba odezvy dialogového kroku.....	30
5.1.3	Transakční profil.....	31
5.1.4	Časový profil.....	33
5.2	NEDOSTATEČNÝ HARDWARE.....	34
5.3	NEROVNÉ ROZLOŽENÍ ZÁTĚŽE.....	35
5.4	MONITOR OPERAČNÍHO SYSTÉMU.....	35
5.5	NÁROČNÝ PROCES.....	37
5.6	MONITOR DISKŮ.....	40
5.7	MONITOR LAN.....	40
5.8	MONITOR ZMĚN PARAMETRŮ.....	41
6	VÝKON DATABÁZE.....	42
6.1	ANALÝZA BUFFERŮ DATABÁZE.....	42
6.2	EXKLUZIVNÍ ZÁMKY.....	43
6.3	ZÁZNAM O STAVU DATABÁZE.....	44
6.4	SQL AKTIVITY TRANSAKCE.....	44
7	ANALÝZA KONFIGURACE PAMĚTI.....	45
7.1	MONITOR KONFIGURACE PAMĚTI.....	45
8	SAP HANA.....	48
8.1	POROVNÁNÍ S KONVENČNÍMI DATABÁZEMI.....	48
8.2	UKLÁDÁNÍ DAT.....	48
8.3	HARDWARE PRO SAP HANA.....	49
8.4	PŘECHOD NA SAP HANA.....	49
II	PRAKTICKÁ ČÁST.....	51
9	OPTIMALIZAČNÍ PROJEKT.....	52
9.1	KLÍČOVÉ UKAZATELE VÝKONNOSTI.....	52
9.2	ZLEPŠENÍ PRŮMĚRNÉ DOBY ODEZVY.....	52
9.3	ZMĚNA PAMĚŤOVÝCH PARAMETRŮ.....	56
9.4	OPTIMÁLNÍ ROZLOŽENÍ ÚLOH.....	60
	ZÁVĚR.....	68
	SEZNAM POUŽITÉ LITERATURY.....	69
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	70
	SEZNAM OBRÁZKŮ.....	71
	SEZNAM TABULEK.....	73

ÚVOD

Při provozování ERP systému jsou výkon a dostupnost velice podstatné. Ovlivňují nejen spokojenost uživatelů, ale v neposlední řadě též produktivitu a výsledky firmy. Pomalý nebo nedostupný systém způsobuje finanční ztráty, zpoždování dodávek, přesčasy zaměstnanců nebo poškození dobrého jména společnosti.

Výkon systému je definován, jako schopnost systému splnit požadavky v rychlosti odpovědi a datové propustnosti. Není to však jen nějaké absolutní číslo například průměrné doby odpovědi na požadavek. Jedná se spíše o relativní pojem závisející na požadavcích na danou aplikaci.

Tato publikace předpokládá některé základní teoretické znalosti a praktické zkušenosti z oblasti administrace systémů SAP. Není zde popisován postup přihlášení do systému či spuštění transakcí. Předpokládáme, že zájemce o analýzu a optimalizace výkonnosti systémů SAP již takovými znalostmi disponuje. Není zde ani rozebrána optimalizace hardwaru či propustnosti sítě, které jsou spíše záležitostmi dané platformy.

I. TEORETICKÁ ČÁST

1 SYSTÉM SAP A TYPY INSTANCÍ

Systémy SAP patří do skupiny ERP (Enterprise Resource Planning) software. Jedná se tedy o systémy pro plánování podnikových zdrojů. Samotná zkratka SAP pochází z německého „Systeme, Anwendungen, Produkte in der Datenverarbeitung“, v překladu do češtiny „Systémy, aplikace, produkty ve zpracování dat“. Software byl vyvinut firmou SAP SE se sídlem v německém městě Walldorf. Pro zajímavost ještě poznamenejme, že do roku 2014 se společnost jmenovala SAP AG.[1]

Existuje mnoho různých typů systémů SAP. Všechny jsou však založené na jednom ze dvou programovacích jazyků. První možností je jazyk ABAP, druhou pak Java. Postup optimalizace se zásadně liší v závislosti na použitém jazyku. V této předkládané práci se budeme zabývat technologií ABAP.[1]

1.1 Vymezení pojmů

Na úvod je třeba osvětlit některé pojmy, které jsou využity hojně v dalším textu. Mnohé z nich jsou převzaty z anglického jazyka a nepřekládají se ani v českých publikacích. Pro lepší pochopení jsou tedy ponechány tak, jak se s nimi čtenář setká v reálném systému. Takovým pojmem je work proces. Takto je označen proces, který vykonává v systému SAP určitou samostatnou funkci. Každý work proces je reprezentován právě jedním procesem operačního systému.[1]

Dále je dobré osvětlit rozdíl mezi vyrovnávací pamětí a mezipamětí. Jsou to dva typy paměti, které se nachází mezi rychlejším a pomalejším zařízením. Typicky například procesor a disk, nebo proces a databáze. Význam je především takový, že rychlejší zařízení není zbytečně zpomalováno čtením z pomalejšího zařízení. Vyrovnávací paměť je typu fronta. Tedy všechna data jsou zpracována postupně. V anglickém jazyce je tato paměť pojmenována „buffer“. Známý je termín „přetečení bufferu“, což značí situaci, kdy se rychlejší zařízení snaží zapsat do paměti více dat, než je velikost této paměti. Naproti tomu mezipaměť je spíše místem pro nejčastěji čtená data. Informace se tak získá z paměti mnohem rychleji než z pomalejšího zařízení. V anglickém jazyce se tato paměť označuje „cache“. Pokud nastane požadavek na uchování dalších dat pro která v cache paměti není místo, některá méně použitá data jsou z paměti odstraněna. Data jsou uložena ještě na jiném místě (disk, databáze), takže ke ztrátě dat nedochází.

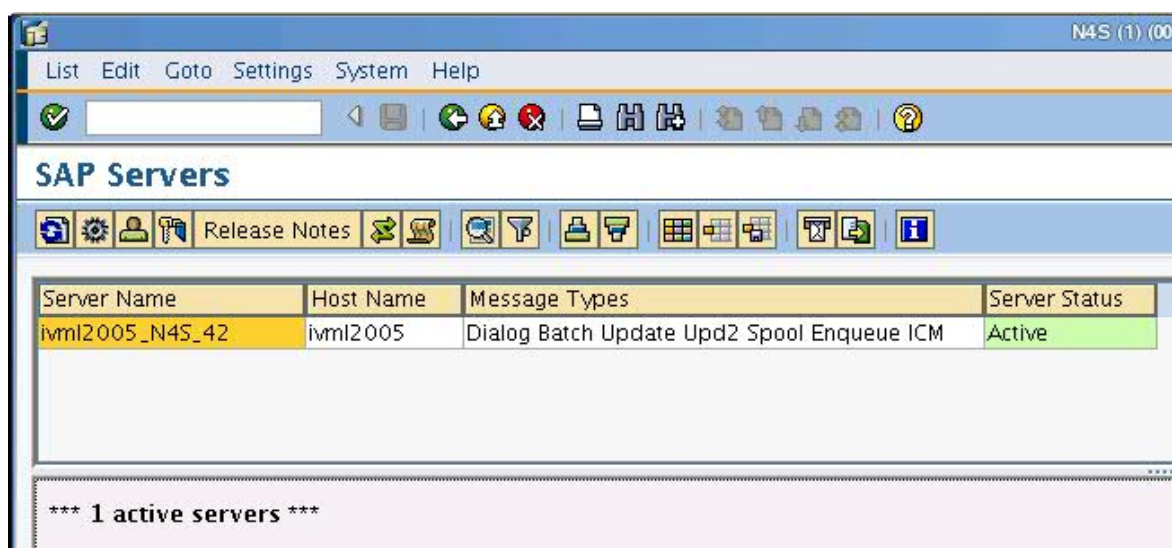
1.2 Vrstvy systému SAP

SAP lze rozdělit do tří vrstev. První je vrstva databázová, následuje aplikační a poslední je prezentační. Databázová vrstva je tvořena databází. Zvyšování výkonu v této vrstvě možné je, ale jedná se o optimalizaci výkonu databáze. To není cílem této publikace. Pro systémy SAP lze použít mnoho databází. Pro zvýšení výkonu v této vrstvě tedy použijeme doporučení pro danou databázi. Nutno poznamenat, že je třeba následovat doporučení společnosti SAP SE (výrobce systému SAP) a nikoliv obecná doporučení vydavatelů databází. Konfigurace a využití databáze pro systém SAP jsou totiž natolik specifické, že se liší od obecných doporučení vydavatelů. Přesto se budeme zabývat některými společnými ukazateli výkonu všech databází obecně a optimalizací parametrů, které jsou použité na straně systému SAP a s databází přímo souvisí.[1]

Aplikační vrstva zpracovává požadavky od uživatele a tvoří jádro aplikace. Zde bude hlavní zaměření naší práce. Prezentační vrstvu tvoří grafické uživatelské rozhraní, které je nainstalováno na straně uživatele. Je třeba pravidelně instalovat nejnovější verze tak, jak výrobce doporučuje. Jedná se o jednoduchou instalaci programu a tato instalace není závislá na konfiguraci systému SAP.[1]

1.3 Instance SAP

Každý systém SAP se skládá z jedné, nebo více instancí. Instance je určitá samostatná jednotka, která může fungovat nezávisle na ostatních. Může být samostatně nastartována, nebo zastavena. Přehled všech instancí systému SAP zobrazíme v transakci SM51 (Obr. 1).



The screenshot shows the SAP transaction SM51 'SAP Servers'. The window title is 'N4S (1) (001)'. The menu bar includes 'List Edit Goto Settings System Help'. Below the menu is a toolbar with various icons. The main area contains a table with the following data:

Server Name	Host Name	Message Types	Server Status
ivml2005_N4S_42	ivml2005	Dialog Batch Update Upd2 Spool Enqueue ICM	Active

At the bottom of the window, the text '*** 1 active servers ***' is displayed.

Obr. 1. Přehled instancí

1.3.1 Centrální instance

Centrální instance obsahuje všechny servery a procesy, které jsou zapotřebí pro běh systému SAP. Pro systémy malého rozsahu je typické, že je systém tvořen pouze centrální instancí, která je nainstalována na stejném serveru jako databáze. Z pohledu vrstev je tedy na jednom místě provozována databázová vrstva a aplikační vrstva. Uživatel se svojí prezentační vrstvou (SAP GUI) je připojen vzdáleně přes síť.[1]

1.3.2 Dialogová instance

Dialogová instance se také někdy označuje jako aplikační instance. Obsahuje procesy označené v této práci jako „ostatní procesy“. Tato instance nemůže být spuštěna samostatně a je vždy navázána na centrální instanci či ASCS. Obsahuje též některé ze základních procesů. [1]

1.3.3 ASCS instance

Zkratka ASCS znamená ABAP SAP Central Services, tedy centrální servery pro SAP systém typu ABAP. Tato instance obsahuje všechny servery označené zde za základní, nikoliv však ty ostatní. Díky nízkému počtu procesů a alokované paměti může být tato instance velice rychle nastartována, což je požadavek při použití clusteru. O tom bude pojednáno později.[1]

1.3.4 ERS instance

Zkratka ERS je složená ze slov Enqueue Replication Server. Hlavní význam instance tohoto typu spočívá v držení informace o zámcích na tabulkách. Tato instance má zásadní význam pro clusterové řešení, protože udržuje informaci i v případě pádu některé z instancí.[1]

1.4 Výběr SAP instancí

V následujících kapitolách budeme uvažovat, kdy je vhodné použít konkrétní typ instance. Budeme se zabývat též vlivem na výkon a v neposlední řadě i dostupnost systému. Výrobce nedoporučuje instalovat na stejný server produkční instance a jiné instance (tedy instanci systému kvality či vývoje).

1.4.1 Jednoduché řešení

Jak již bylo zmíněno výše, nejjednodušší možností je použití centrální instance. Je to typická konfigurace pro málo vytížené produkční systémy nebo pro systémy ověření kvality (Quality Assurance) a vývoje (Development). Toto řešení přináší řadu výhod v podobě jednoduššího nastavení parametrů a snadnější administrace.

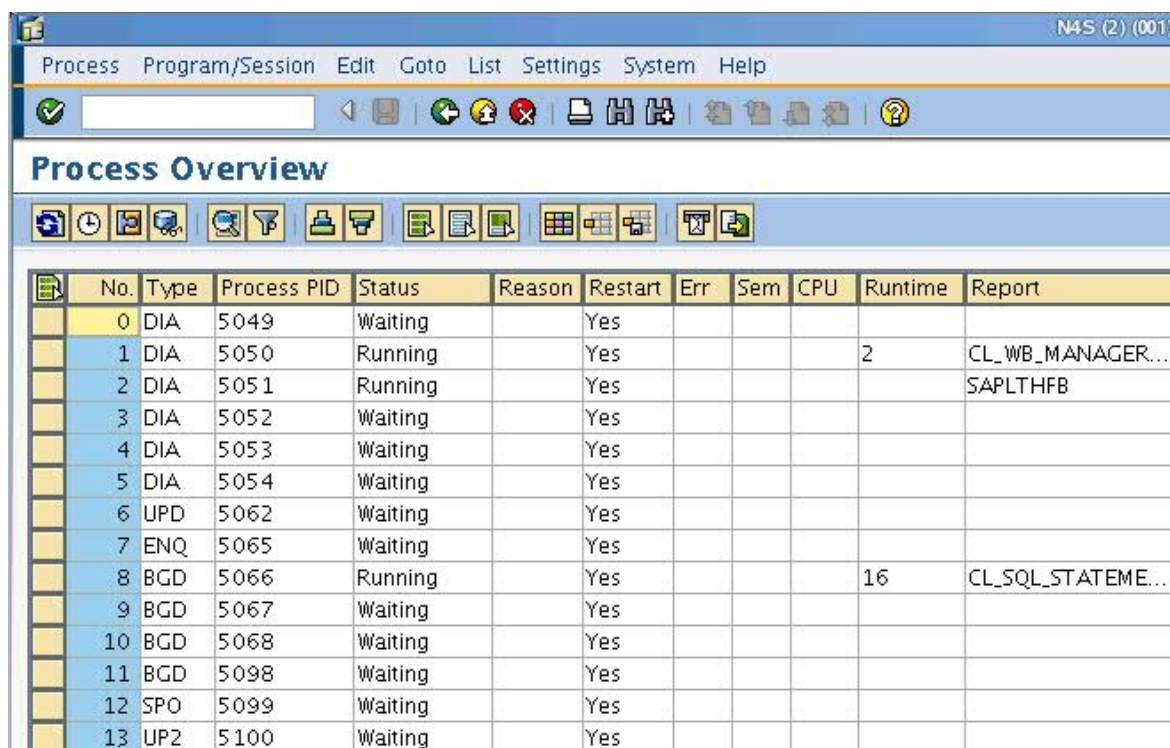
1.4.2 Více výkonu

Pro zvýšení výkonu je jednou z možností přidání dialogové instance. To zahrnuje samozřejmě nákup dalšího hardwaru, zapojení do stávající infrastruktury a následnou instalaci dialogové instance samotné. Získáme tak více výpočetního výkonu pro náš SAP systém. Takto můžeme přidat velký počet dialogových instancí. Jediným omezením (kromě finančního hlediska za hardware) bude konečný počet možností pro výběr čísla instance. Toto je v rámci jednoho SAP systému unikátní dvouciferné číslo.[1]

Teoreticky máme tedy 100 možností. V praxi nelze použít číslo 99, které je vyhrazeno pro SAProuter a též se nedoporučuje použití čísla 01, protože pak může nastat kolize s defaultními porty některých operačních systémů. Dlužno tedy dodat, že čísla portů služeb jsou automaticky generována z příslušného čísla instance. Kupříkladu dispatcher má číslo portu 32<číslo_instance>. Pro instanci s číslem 15 by tedy bylo číslo portu dispatcheru 3215.[1]

1.5 Zvýšení výkonu instance

Další možností, jak zvýšit výkon, je navýšení počtu jednotlivých procesů instance, za předpokladu, že adekvátně posílíme i hardware. Zde máme možnost navýšit pouze některé procesy, a cíleně tak zvýšit výkon v určité oblasti. Následně probereme jednotlivé procesy a situace, které signalizují potřebu jejich zvýšení. Tato možnost přichází v úvahu pouze pro centrální instanci a aplikační instanci. Ostatní instance neobsluhují požadavky uživatele. Neobsahují totiž potřebné procesy. Přehled procesů instance zobrazíme prostřednictvím transakčního kódu SM50 (Obr. 2).[1]



The screenshot shows the SAP Process Overview window. The title bar indicates 'N4S (2) (001)'. The menu bar includes 'Process', 'Program/Session', 'Edit', 'Goto', 'List', 'Settings', 'System', and 'Help'. Below the menu bar is a toolbar with various icons. The main area displays a table with the following data:

No.	Type	Process PID	Status	Reason	Restart	Err	Sem	CPU	Runtime	Report
0	DIA	5049	Waiting		Yes					
1	DIA	5050	Running		Yes				2	CL_WB_MANAGER...
2	DIA	5051	Running		Yes					SAPLTHFB
3	DIA	5052	Waiting		Yes					
4	DIA	5053	Waiting		Yes					
5	DIA	5054	Waiting		Yes					
6	UPD	5062	Waiting		Yes					
7	ENQ	5065	Waiting		Yes					
8	BGD	5066	Running		Yes				16	CL_SQL_STATEME...
9	BGD	5067	Waiting		Yes					
10	BGD	5068	Waiting		Yes					
11	BGD	5098	Waiting		Yes					
12	SPO	5099	Waiting		Yes					
13	UP2	5100	Waiting		Yes					

Obr. 2. Přehled work procesů

1.5.1 Dialog proces

Dialog proces slouží k obsluze požadavků uživatele. Ten využívá dialog proces pouze v době od odeslání požadavku do jeho přijetí. Typicky kolem jedné vteřiny. Jeden dialog proces je tak schopen obsloužit postupně více uživatelů. Pokud není volný žádný dialog proces, tak uživatel čeká. Zdali tento jev nastává, lze zjistit v transakci ST03. Počet dialog procesů navýšíme změnou parametru instance. Daný parametr nese název `rdisp/wp_no_dia.[1]`

1.5.2 Background proces

Tento proces slouží ke spuštění úloh v systému SAP (anglicky batch jobs). Opět platí, že jeden proces obsluhuje právě jednu úlohu. Proces je blokován úlohou až do doby, než úloha skončí. To může trvat od jednotek vteřin až třeba po několik dní, většinou jsou to ale minuty či desítky minut, výjimečně pak hodiny. Pokud není žádný volný background proces ve chvíli, kdy daná úloha dosáhne času spuštění, nastává stejná situace jako v předchozím případě u dialogových procesů. Úloha musí čekat. V případě jednotlivých úloh lze definovat nejzazší možný čas, kdy se má úloha spustit. Jestliže nedojde ke spuštění do

daného okamžiku, úloha se již nespustí. Zpoždění či vynechání úloh lze kontrolovat přes transakci SM37. Počet background procesů navýšíme změnou parametru rdisp/wp_no_btc. [1]

1.5.3 Spool proces

Spool proces slouží k obsluze tiskových úloh. V praxi nebývá zapotřebí navyšovat počet těchto procesů. Z našich zkušeností vyplývá, že pokud společnost tiskne i mnoho tisíc výtisků za den, tak jediný spool proces plně postačí. Limit je spíše na straně tiskáren. Příslušný parametr se jmenuje rdisp/wp_no_spo.[1]

1.5.4 Update proces

Pokud dialogový proces požádá o asynchronní update na databázi, je tento požadavek předán update procesu. Tento proces se též stará o zpracování požadavků na update z jiných systémů. Přehled požadavků na update je zobrazitelný přes transakci SM13. Pokud se vyskytuje více požadavků ve stavu „čekající“, je nutno bližší posouzení důvodu. Pakliže je čekání způsobeno nedostatkem volných update procesů, je třeba navýšit parametr rdisp/wp_no_ub.[1]

1.6 Operační módy

Majorita uživatelů ve společnosti bez směnného provozu využívá SAP systém nejvíce v pracovních hodinách od pondělí do pátku. Nabízí se otázka, zdali nakonfigurovat systém tak, aby v této době systém poskytoval více výkonu pro uživatele, zatímco ve zbylém čase byl výkon dostupný více pro úlohy na pozadí. Řešením jsou operační módy. Příslušný transakční kód je RZ04 (Obr. 3). Je třeba nastavit dva údaje. V prvním kroku je nutné definovat časové úseky jednotlivých módů (obvykle se používá rozdělení denní a noční), ve druhém kroku se definují vlastnosti operačních módů. Lze určit, kolik dialogových procesů se má změnit na background procesy. Tímto způsobem lze přesunout výpočetní výkon mezi uživateli a úlohami na pozadí.[1]

2 CLUSTER

Pro mnohé společnosti je dostupnost jejich SAP systému téměř životně důležitá. Může na ní záviset výroba, komunikace se zákazníky, objednávkový systém či například organizace logistiky. Případný výpadek má za následek velké finanční ztráty nebo i poškození dobrého jména firmy. Je logické, že se společnosti snaží tato rizika minimalizovat.

2.1 Základní funkce

Uveďme modelovou situaci, kdy nastane porucha systému SAP, který je nainstalován pouze na jednom serveru bez použití clusteru. V následujícím momentu uživatelé kontaktují správce systému. Správce začne v ideálním případě okamžitě na problému pracovat a systém opraví. Přesto bude systém nedostupný desítky minut.

Může nastat ale i situace, že se vyskytne chyba na straně hardwaru a náprava může vyžadovat koupi nové části serveru (napájení, chlazení). Doba opětovného zprovoznění systému pak může činit i dny. Takovému scénáři lze předejít použitím clusteru.

Cluster je tvořen dvěma servery a je nakonfigurován tak, že (SAP) systém je spuštěn pouze na jednom ze serverů. Pokud nastane problém, ať už hardwarový nebo softwarový, tak se běh systému ukončí a automaticky se nashoduje na serveru druhém. Tím se značně minimalizuje doba nedostupnosti systému. Pokud nastane porucha hardwaru jako v předchozím případě, systém se nashoduje na druhém serveru a doba nedostupnosti se bude počítat v minutách.[2]

V případě clusteru je výhodné využití ASCS instance právě z důvodu minimálního počtu procesů. Taková instance se při potížích rychleji ukončí na problémovém serveru a též rychleji nashoduje na druhém serveru. Doba nedostupnosti systému v tomto případě může být pouze několik desítek vteřin. Samozřejmě musí být nainstalována minimálně jedna dialogová instance ještě na jiném serveru pro obsluhu požadavků.

2.2 Synchronizace souborového systému

Souborový systém může být umístěn na lokálním disku serveru. V tuto chvíli nastává otázka, jak je řešena synchronizace stavu souborového systému při přepnutí clusteru. Je zde potřeba rozlišit soubory jednotlivých vrstev (databázové a aplikační). V případě aplikační vrstvy problém nenastává, soubory se s během systému nemění. Jsou tedy stejné na obou serverech. Jinak je tomu u databází. Informace jsou uloženy v souborech, které se neustále

mění a při pádu systému potřebujeme mít stejné soubory i na druhém serveru, abychom nepřišli o data. Toho můžeme docílit tak, že na neaktivním serveru nastartujeme databázi do stavu „recovery“ a budeme tam aplikovat logy (změny) z aktivního serveru. Všechny provedené změny se budou replikovat a databáze na neaktivním serveru bude stále připravená pro případný start se stejnými daty, jako jsou na aktivním serveru.[2][3]

Pokud máme disk připojený přes síť (například SAN – Storage Area Network), situace je jednodušší. Pak se může disk jednoduše přepojit na kterýkoliv server současně s přesunem instance.[3]

2.3 Využití neaktivních serverů

Může vzniknout námitka, že neaktivní server je zcela zbytečně nevytížený, a to je plýtvání. Vezměme v úvahu, že aktivní i neaktivní server musí mít stejnou hardwarovou konfiguraci. Jen tak lze zajistit, že při přepnutí bude software fungovat stejně. Jak tedy neaktivní server využít a přitom neohrozit dostupnost systému? Možností je několik. Kupříkladu můžeme na neaktivní server nainstalovat dialogovou instanci. Tímto zvýšíme výkon SAP systému a využijeme oba servery. Při problémech na aktivním serveru je tato dialogová instance ukončena a na jejím místě se nastartuje centrální instance (případně ASCS).[2]

Další možnou konfigurací je umístění centrální instance na jeden server a databáze na druhý. Zde již nelze hovořit o aktivním a neaktivním serveru, neboť jsou oba aktivní. Při chybě na jednom serveru dojde k přesunu běžícího softwaru, ať už databáze, nebo centrální instance, na druhý server. SAP systém je po té opět dostupný, avšak s menším výkonem oproti původnímu stavu. Nicméně tato konfigurace přináší jedno velké nebezpečí. Může se totiž stát, že při špatně zvolených parametrech (zvláště paměťových) druhý server nebude mít dostatek paměti a centrální instance na druhém serveru nenastartuje. Proto tuto konfiguraci nedoporučujeme.[2]

2.4 Cluster bez investic do hardwaru

Požadavek vysoké dostupnosti v podobě clusteru bez další investice do hardwaru se zdá neřešitelný. Jedna možnost však existuje. Pro každý produkční systém je nainstalován i systém kvality a systém vývoje. Zatímco vývojový systém je svým rozsahem a též hardwarem poměrně limitovaný, systém kvality je velice podobný produkčnímu systému, a to z níže popsaných důvodů.

Funkce systému kvality je testování různých změn ještě před uvedením do produkčního systému. Když nové změny vedou například k nestabilitě nebo pádu, problém je včas odhalen a změny nejsou do produkčního systému aplikovány. Proto systém kvality musí mít velice podobnou (ideálně stejnou) konfiguraci jako produkční systém. A zde přichází možnost využití pro cluster.

Pro každou společnost bývá nejdůležitější dostupnost produkčního systému, zatímco výpadek systému kvality není tolik podstatný. Proto lze nakonfigurovat cluster produkčního systému takovým způsobem, že při chybě je ukončen běh systému kvality a na jeho místě je nastartován systém produkční.

2.5 Síťová konfigurace clusteru

Vyvstává nám nyní otázka, na který server se má uživatel připojit, když se SAP systém může vyskytnout na kterémkoliv ze serverů, které jsou do clusteru zapojené. Funkcionalita clusteru to řeší zavedením virtuálních IP adres a též virtuálních jmen serverů. Tyto virtuální jména a adresy ukazují vždy na aktivní server, a proto se uživatel vždy připojí na server, který je aktivní. Nutno ještě podotknout, že už při instalaci SAP systému je zapotřebí tyto virtuální jména (adresy) použít. Nelze tedy překonfigurovat již existující základní řešení. V takovém případě je třeba znovu nainstalovat SAP systém a nahrát všechna data jedním z podporovaných řešení. Nejjednodušeji zálohou obsahu databáze a následnou obnovou do nového řešení.[2]

3 PAMĚŤ SYSTÉMU SAP

Doposud jsme se zabývali pouze zvyšováním výkonu za cenu navýšení hardwaru a úpravou konfigurace systému. V následujících kapitolách se však zaměříme na parametry a nastavení systému, které výkon přímo ovlivňují. V tomto aspektu jsou jedny z nejdůležitějších paměťové parametry. Paměť může být sdílená, nebo lokální. Sdílená paměť je dostupná všem procesům systému, lokální pouze jednomu danému. Systém SAP nemůže přímo alokovat fyzickou paměť serveru. Alokace probíhá vždy ve virtuální paměti. Operační systém pak rozhodne, zdali bude obsazena fyzická paměť, nebo odkládací prostor (swap).[4]

První paměť alokovaná při obsluze požadavku je roll memory. Zde se nahraje tzv. „kontext“ uživatele, který požadavek vytvořil. Details jsou popsány v jiných kapitolách. Další alokovanou pamětí je extended memory. Tato paměť je základní a nejdůležitější pro práci systému SAP. Každý work proces má rezervovanou část adresového prostoru v extended memory. Paměť je přiřazována dynamicky podle potřeby. V případě, že work proces vyčerpá roll memory a extended memory, pak dochází k alokování private memory. Někdy je tato paměť označována též jako heap memory. Při alokování této paměti work proces přechází do tzv. PRIV módu. To má za následek blokování work procesu po celou dobu transakce, což může být problém.[4]

Představme si situaci, kdy uživatel spustí transakci, která ukládá do paměti velký objem dat. Work proces se přepne do PRIV módu a není dostupný pro ostatní uživatele. Pokud je takových požadavků od uživatele více nebo mnoho uživatelů spustí takový požadavek, může dojít k obsazení všech volných work procesů. To je vnímáno ostatními uživateli jako nedostupnost systému. Pokud pro uživatele neexistuje volný work proces, jeho požadavky nejsou obsluhovány. Vzniklou situaci lze řešit dvěma způsoby. Buď můžeme vyčkat na dokončení úlohy work procesů. Nelze však odhadnout, jak dlouhá bude doba čekání. Druhou možností je přihlášení přímo na operační systém a restartování některého z work procesů. Následně lze se systémem opět pracovat a zahájit kroky k nápravě situace. Uživatel, který využíval daný work proces, však ztratí data. Nicméně se stále jedná o přijatelnější řešení než restartovat celý systém, nebo odevzdaně čekat, zdali bude systém dostupný později.[4]

3.1 Paměťové parametry

V této kapitole je uveden přehled parametrů, které se týkají paměti vyhrazené pro work procesy. Kterýkoliv parametr v systému SAP lze zobrazit v transakci RZ10. Pro lepší přehlednost jsou parametry uvedeny v odrážkách a za každým parametrem vždy následuje jeho vysvětlení.

- abap/heaplimit

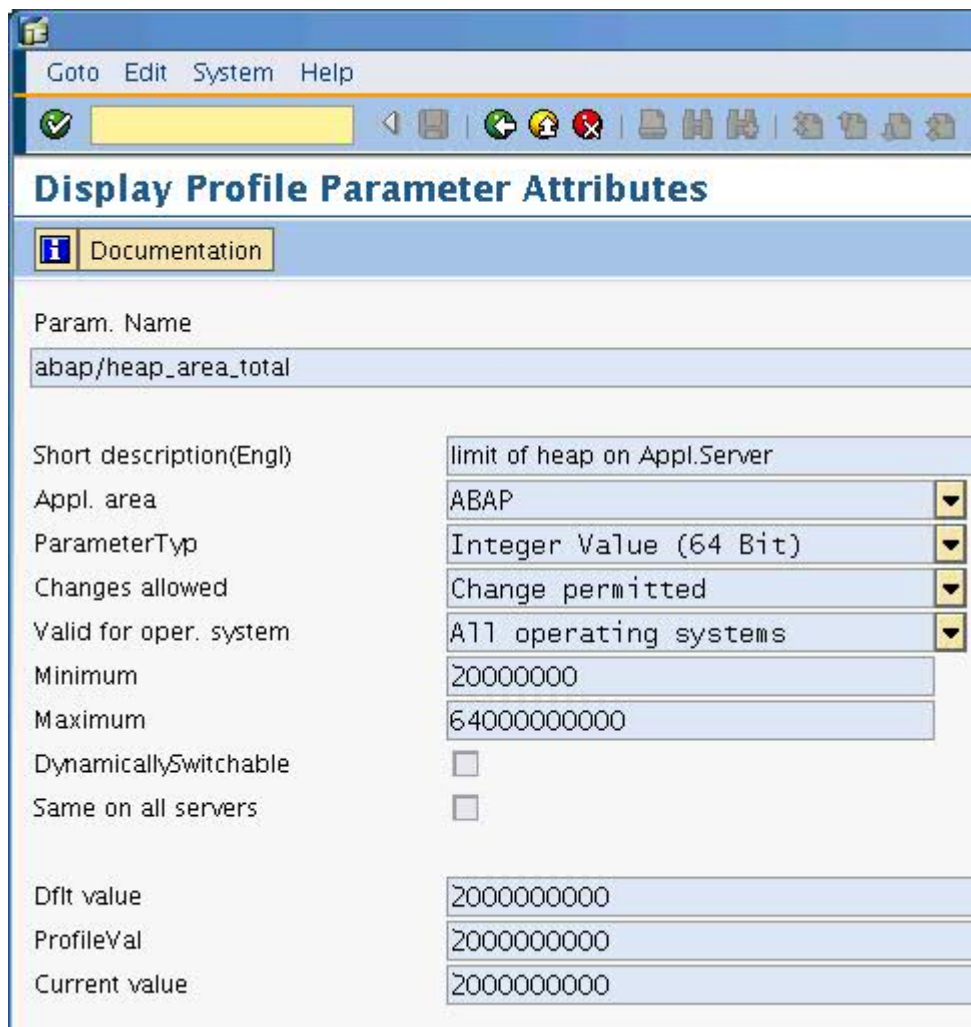
Představme si situaci, kdy uživatel spustí náročný požadavek. V důsledku toho přiřazený work proces alokuje velké množství paměti. Po obslužení daného požadavku zůstává paměť stále alokována. Jsou zde ukládána data dalšího požadavku. Pokud paměť alokovaná jedním work procesem přesáhne určitou mez, tak po skončení dialogového kroku nastane restart tohoto work procesu. Bylo by zbytečné, aby měl work proces alokovan hodně paměti a přitom se využívala jen malá část. Tato mez pro restart je právě dána parametrem abap/heaplimit.[5]

- ztta/roll_extension
- ztta/roll_extension_dia
- ztta/roll_extension_nondia

Předně je třeba osvětlit pojem kontext uživatele (anglicky user context). Jedná se o data uživatele, která si work proces musí načíst před tím, než začne požadavek uživatele obsluhovat. Tato data obsahují například přístupová práva, výsledky předchozích kroků nebo informace uživatele. Parametr ztta/roll_extension definuje maximální množství paměti, které může být alokováno jedním work procesem pro kontext z oblasti extended memory. Toto omezení lze ještě blíže specifikovat podle skutečnosti, zdali se jedná o dialogový nebo jiný work proces. V takovém případě se nastavují parametry ztta/roll_extension_dia respektive ztta/roll_extension_nondia.[5]

- abap/heap_area_total

Parametr abap/heap_area_total definuje maximální množství heap memory pro všechny work procesy v systému. Základní hodnota je závislá na platformě systému.[5]



Obr. 3. RZ11 – parametr abap/heap_area_total

- abap/heap_area_dia
- abap/heap_area_nondia

Výše uvedené parametry udávají maximální množství paměti, které může být alokováno jedním dialogovým (respektive nedialogovým) work procesem v heap memory. Pokud alokace work procesu dosáhne této hodnoty, tak je work proces restartován. Tímto je zajištěno, že jediný uživatel v systému nemůže obsadit velké množství paměti a způsobit tak výkonové problémy.[1][5]

- ztta/roll_area

Tento parametr slouží pro definování velikosti roll memory. Jak je popsáno výše, tato paměť je využívána jako první při obsluze požadavku.[5]

- `em/initial_size_MB`

Parametr `em/initial_size_MB` určuje velikost extended memory. Tento parametr musí být vždy větší než parametr `ztta/roll_extension`. [5]

- `rdisp/ROLL_SHM`

Roll buffer slouží k urychlení načtení kontextu. Ostatní parametry obvykle udávají hodnoty v bitech případně MB, avšak `rdisp/ROLL_SHM` značí počet 8 KB bloků. Na to je třeba brát zřetel při zadávání. Pro nastavení velikosti bufferu můžeme počítat přibližně 1 MB pro každého uživatele. [5]

4 OPTIMALIZACE

Optimalizace výkonu systému je proces, který vždy obsahuje 5 fází:

- Porozumění business procesům firmy
- Nastavení výkonnostních cílů
- Systematické monitorování, identifikace a analýza problémů
- Implementace řešení
- Analýza (ověření) výsledků optimalizace [4]

Obecně lze uvést ještě několik doporučení:

- Analyzovat komplexně: Je důležité provádět analýzu různých parametrů a chování systému, abychom správně identifikovali příčinu problému.
- Neprovádět změny bez analýzy: Při opakovaném problému na systému, případně stejných symptomech, známých z jiného systému, může mnoho administrátorů napadnout myšlenka, že aplikují stejné řešení, které bylo použito k vyřešení předchozího problému. Toto je ovšem velice nebezpečné a jakákoliv změna by měla být provedena pouze po předchozí důkladné analýze.
- Změny provádět po malých krocích: Úspěch optimalizace závisí na správných hodnotách a těch se lépe docílí opakovaným přenastavováním po menších krocích.
- Neprovádět příliš mnoho změn: Tímto máme na mysli příliš mnoho změn najednou. Snadno by se pak mohlo stát, že změny některých parametrů výkonnost systému zvýší a jiné ho naopak sníží. V celkovém součtu pak rozdíl může být neznatelný a můžeme nabýt dojmu, že dané změny nemají na výkon vůbec žádný vliv.
- Každé pravidlo má výjimku: Každý systém je unikátní a jistě se časem setkáme s doporučením, které je obecně správné, nehodí se však pro náš systém. Nebo naopak obecně nevhodné, pro náš systém však správné.[4]

4.1 Technická optimalizace

Z technického hlediska je aplikace složena z mnoha různých komponent. Na jedné straně jsou to logické komponenty jako například procesy, vlákna, servisy, vyhrazená paměť (například buffer), na druhé straně jsou zde fyzické komponenty jako CPU, RAM, disky

nebo síťové prvky. Všechny tyto komponenty musí být vyvážené. Pokud je některá z komponent nedostatečně dimenzovaná, stane se úzkým místem (anglicky bottleneck). Jediná komponenta pak výrazně negativně ovlivní výkon celého systému.[4]

4.2 Aplikační optimalizace

Dalším důležitým bodem je optimalizování běhu systému z aplikačního pohledu. Je tím myšleno například identifikace úloh, které nejsou potřebné, ale přesto jsou v systému naplánovány. Též sem patří neefektivní programy, případně neefektivní využívání programů, byť efektivních.[4]

Pokud ani po technické a aplikační optimalizaci systém nevykazuje žádanou výkonnost, je třeba navýšit stávající zdroje komplexně.

4.3 Jak často optimalizovat

Odpověď na tuto otázku není jednoznačná a záleží na několika aspektech. Prvním z nich je typ systému. Již dříve bylo zmíněno, že existují tři typy systémů: produkční systém, testovací systém, nebo též systém ověření kvality, a vývojový systém. Největší význam má optimalizování produkčního systému, protože ho používá nejvíce uživatelů. Špatný výkon produkčního systému má zpravidla největší dopady. Systém vývoje není tolik důležitý. Na něm obvykle pracuje pouze několik uživatelů, kteří vyvíjejí nové programové funkce či upravují stávající objekty. Navrhované změny testují na systému ověření kvality, který je obvykle pro firmu nejméně důležitý.[4]

Dalším aspektem pro posouzení četnosti optimalizací je velikost systému. Dá se předpokládat, že na velkém a vytiženém systému bude mnohem více prostoru k optimalizaci než na systému malém nebo nevytiženém. Je vhodné též provést optimalizaci po velkých změnách na systému, jako jsou například přechod na novější verzi nebo při nahrání velkého objemu dat do systému. V neposlední řadě se provádí optimalizace před prvním spuštěním systému (po instalaci) a následně po zahájení provozu. Samozřejmě provádíme optimalizaci vždy, když pocítujeme nedostatečný výkon aplikace.[4]

4.4 Zdroj problémů

Mnoho problémů je způsobeno zásahem do standardních objektů systémů SAP. Takový zásah by měl být velice dobře zvážen. Tvorba neoptimalizovaného kódu způsobuje rovněž

velké množství problémů. A v neposlední řadě musíme zmínit též nedostatečné testování změn, které jsou aplikovány do produkčního systému.[4]

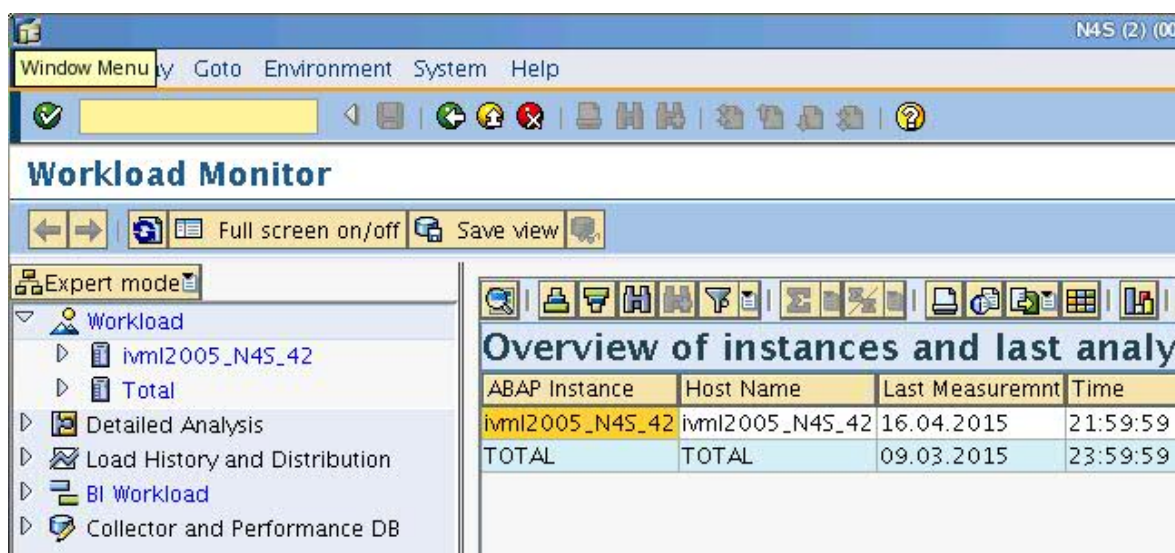
Častým problémem jsou náročné SQL příkazy, které zvyšují zátěž databáze a vedou k prodloužení doby odezvy na požadavek, a nebo ke zbytečným investicím do HW. Udržování velkého objemu dat v systému má velký vliv na zhoršení výkonu. Každý dotaz generuje více dat, záloha databáze trvá delší dobu a paměťové nároky jsou úměrně vyšší.

5 ANALÝZA ZÁTĚŽE

Analýza zátěže poskytuje důležitá data o chování systému a jeho komponent. Dokáže určit vliv jednotlivých transakcí a programů na celkovou zátěž systému. Pomáhá při ujasnění priorit optimalizace výkonu. Nalezená problematická místa je třeba analyzovat ještě podrobněji. Analýza zátěže je startovní bod pro aplikační analýzu.

5.1 Monitor zátěže

Pro každou spuštěnou transakci v systému SAP jsou zaznamenávány statistiky. Jsou ukládány informace například o použití paměti, přístupu do databáze nebo době trvání transakce. Tyto informace zobrazuje Monitor zátěže. Příslušný transakční kód je ST03 (Obr. 4).[4]



The screenshot displays the SAP Workload Monitor interface. The main window title is 'Workload Monitor'. Below the title bar, there is a menu bar with 'Window Menu', 'Goto', 'Environment', 'System', and 'Help'. A toolbar contains various icons for navigation and actions. The interface is divided into several sections:

- Expert mode:** A sidebar on the left with a tree view showing 'Workload' expanded to 'ivml2005_N45_42' and 'Total'. Other options include 'Detailed Analysis', 'Load History and Distribution', 'BI Workload', and 'Collector and Performance DB'.
- Overview of instances and last analysis:** A table on the right showing performance data for different ABAP instances.

ABAP Instance	Host Name	Last Measurement	Time
ivml2005_N45_42	ivml2005_N45_42	16.04.2015	21:59:59
TOTAL	TOTAL	09.03.2015	23:59:59

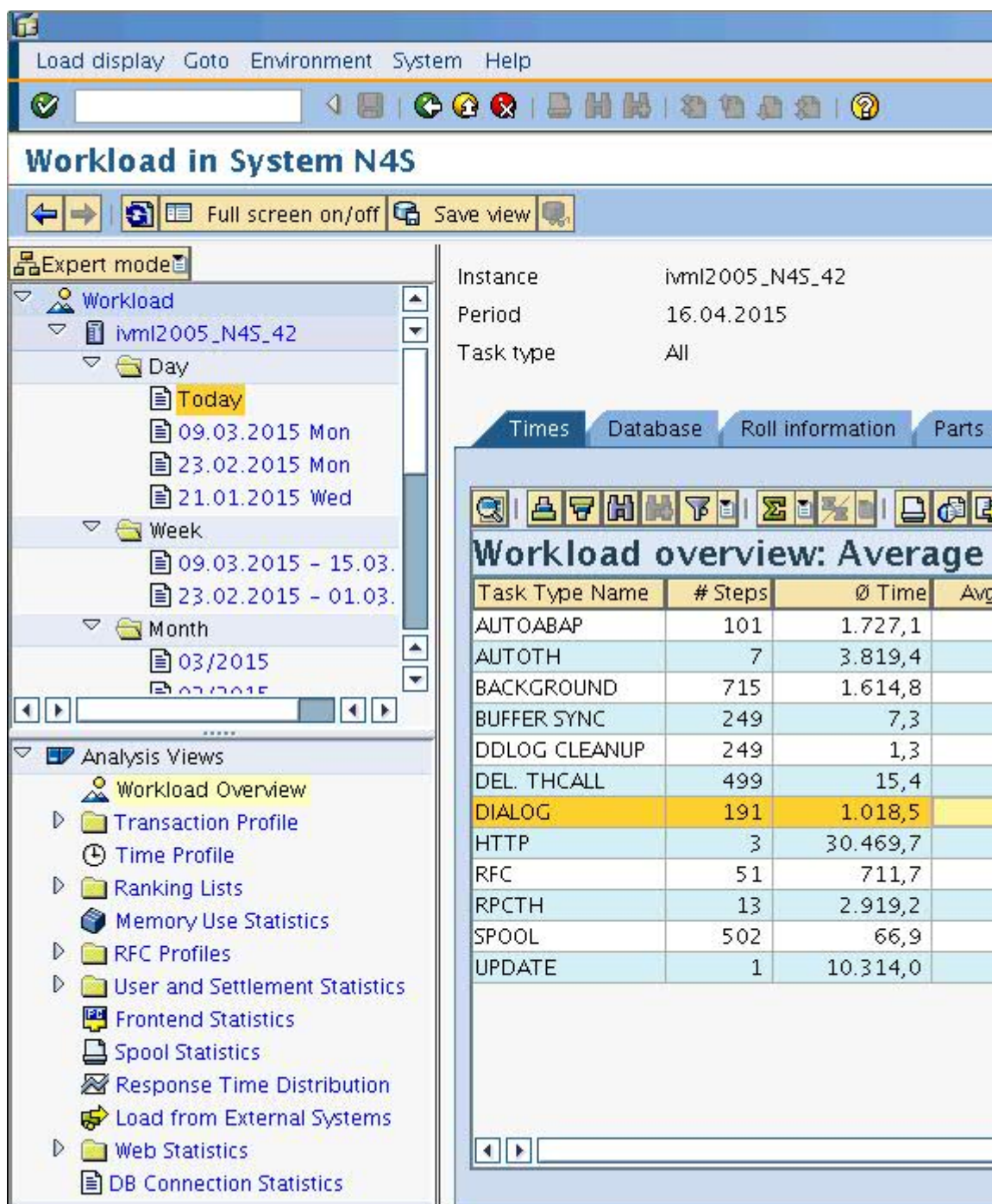
Obr. 4. Transakce ST03

Na výběr jsou tři módy zobrazení: „Administrator“, „Service Engineer“ a „Expert“. První ze jmenovaných je základní uživatelský mód, který zobrazuje aktuální vytížení a statistiky. Rovněž jsou zde dostupné funkce pro servis, který se stará o sběr dat. Druhý mód poskytuje informace o statistikách z předešlého týdne a přehled historie statistik pro všechny aplikační servery. Mód Expert poskytuje přístup ke všem funkcím a informacím, které mohou být využity v transakci ST03. Pro úplnost ještě uvedme, že starší verze systému SAP používaly transakční kód ST03, který zobrazoval poněkud omezenější informace než dnes. V novějších verzích systému byla zavedena transakce ST03N, která tyto informace

rozšiřovala. V současné době se již znovu používá transakční kód ST03, zobrazuje stejné informace jako ST03N.[4]

Předpokládejme situaci, kdy jsme spustili transakci ST03 a vybrali některý mód. Pokud systém SAP obsahuje více instancí, je třeba ještě zvolit určitou instanci pro naši analýzu. Pokud bychom chtěli analyzovat celý systém SAP komplexně, můžeme vybrat volbu Total.

Dále máme na výběr několik různých typů analýz. První a pravděpodobně nejdůležitější je přehled zátěže (workload overview). Tato volba nám zobrazí informace o době odezvy různých procesů v systému. Ve většině případů tyto procesy korespondují s typy work procesů systému SAP. Další sloupec nám značí počet uskutečněných kroků v systému a průměrnou dobu odezvy. Například v dialogové úloze znamená transakční krok jednu změnu obrazovky uživatele. Tedy odeslání požadavku uživatelem, vyřízení požadavku aplikací a následné zobrazení výsledku. Označení „dialogová úloha“ může být poněkud zavádějící, protože sem spadají též některé části úloh, které jsou prováděné na pozadí (background jobs) a také požadavky na update (update request) a tisk (spool request).[4]



Obr. 5. ST03 – Workload Overview

5.1.1 Průměrná doba odezvy

Průměrná doba odezvy (average response time) je většinou označována jako nejdůležitější ukazatel výkonnosti systému. Zvláště citlivě je pak vnímána průměrná doba odezvy dialogového kroku. Zatímco uživatel tolik nepocítuje vyšší průměrnou odezvu úloh na pozadí, v případě odezvy dialogového kroku je tato doba posuzována velmi citlivě, neboť

uživatel čeká před monitorem na zobrazení výsledku své transakce. Za těchto okolností očekává uživatel průměrnou dobu kolem jedné vteřiny, aby odezvu systému hodnotil jako dostatečně rychlou a systém výkonný. SAP SE specifikuje, že výkonnost systému SAP je dostatečná, pokud je průměrná doba odezvy do 1,2 vteřiny.[1] [4]

5.1.2 Doba odezvy dialogového kroku

Rychlost odezvy na požadavek uživatele je reprezentována pojmem "dialog response time". Značí to dobu, za kterou je daný požadavek zpracován a uživatel dostane odpověď. Časově je doba vymezena od přijetí požadavku aplikační vrstvou až do dostupnosti odpovědi pro uživatele, tedy prezentační vrstvou. Čas se tak počítá pouze na aplikační vrstvě. Jedná se o součet časů několika po sobě následujících událostí.[4]

První z událostí je uložení požadavku do fronty požadavků, kde požadavek čeká na volný work proces. Tento čas by měl být velice krátký, řádově milisekundy. V systému SAP se tento čas označuje jako wait time. Pokud je tento čas vyšší než jednotky milisekund, je třeba zkontrolovat volné dialogové work procesy (transakce SM50). Většinou to značí skutečnost, že nejsou žádné dialogové work procesy volné, a je nutné přidat nové.[4]

Pokud požadavek uživatele dostane přiřazen work proces, je zapotřebí nahrát kontext pro daného uživatele. Čas k tomu potřebný je označován jako roll-in time. Tento čas je též poměrně nízký a v praxi zde nevznikají žádné problémy.[4]

Další součástí doby odezvy je čas potřebný pro nahrání kódu programu. Ideálně se kód nachází již přeložený v paměti. Pokud tomu tak není, kód se musí přečíst z databáze a přeložit, a tak se čas významně zvýší právě z důvodu čtení z databáze. Řešením je zvýšení program bufferu (parametr abap/buffersize). Tento čas je označován jako load and generation time.[4]

Největší část výsledného času zaujímá processing time, tedy čas zpracování. Požadavek se zpracovává procesorem serveru (CPU). Tento čas se počítá i tehdy, když je procesor zaneprázdněný jinou úlohou a požadavek zrovna neobsluhuje. Hodnota by měla být blízká CPU time. V opačném případě musíme zjistit, zdali má příslušný server dost výpočetního výkonu.[4]

Lock time, nebo též enqueue time je čas nezbytný k zamknutí SAP objektů. Tato součást dialog response time je obvykle velice nízká, řádově jednotky milisekund.[4]

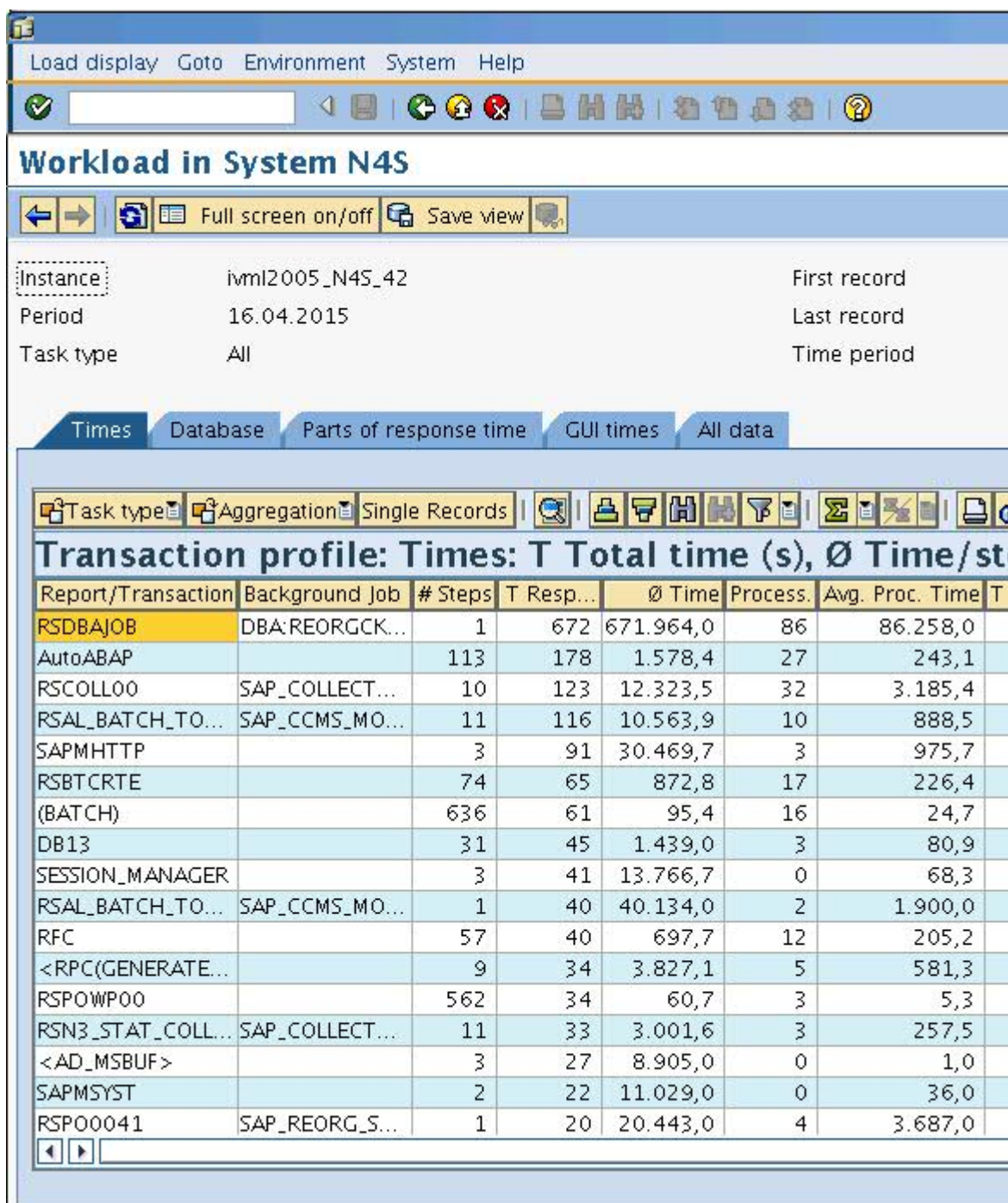
Poslední součástí dialog response time je čas nutný k nahrání zpracovávaných dat z databáze. Tento čas je označován jako database request time. Jestliže je databáze na jiném serveru než daná instance SAP, může se významně projevit i rychlost síťového spojení. Tento čas je počítán na straně instance SAP. Měření začíná, když work proces pošle požadavek na databázové rozhraní a končí ve chvíli, kdy z rozhraní přijde odpověď.[4]

Všimněme si, že do dialog response time není započítáván čas potřebný k přenesení odpovědi po síti. V určitých případech může uživatel vnímat odezvu systému jako špatnou, nicméně problém může být v rychlosti jeho připojení do systému. K tomu dochází například při připojení uživatele z místa mimo kancelář při využití Internetu pro spojení.

5.1.3 Transakční profil

V předchozích dvou kapitolách jsme se zabývali průměrnou dobou odezvy systému. Musíme si uvědomit, že toto číslo je průměrnou hodnotou. Existuje zde nebezpečí, že jediná špatně naprogramovaná transakce výrazně zvýší tento průměr, a přestože systém jako celek bude reagovat velmi rychle, průměrná doba odezvy bude vysoká. Pro pokračování naší analýzy využijeme v transakci ST03 položku „Transaction profile“. Tímto způsobem si lze zobrazit statistiky pro jednotlivé transakce či programy.[4]

Měli bychom se zaměřit pouze na transakce, které byly spuštěné mnohokrát. Představme si kupříkladu transakci, která má velice vysokou dobu odezvy, ale byla spuštěna pouze jednou. Taková transakce nezasáhne do celkového průměru téměř vůbec. Pokud bychom takovou transakci optimalizovali, zabralo by to nepoměrně více času, než by se ušetřilo při dalším ojedinělém spuštění. A v neposlední řadě větší doba odezvy může být způsobena skutečností, že transakce byla spuštěna poprvé, a proto v paměti (bufferech) nebyla ještě nahrána potřebná data. Tak či tak, transakce s nízkým počtem spuštění nejsou vhodné k optimalizaci nebo další analýze.



Obr. 6. ST03 – Transakční profil

Zaměříme se tedy na transakce s velkým počtem spuštění a vyšší dobou odezvy. Nastávají nám dvě možnosti: transakce je standardní transakcí systému SAP, nebo je transakce naprogramována uživatelem. Na první pohled je tento rozdíl zjištělný z transakčního kódu. První písmeno ve jméně takové transakce musí být „Z“. Toto vyžaduje jmenná konvence. V případě standardní transakce bychom měli zkusit vyhledat informace o problému na stránkách výrobce. Může se stát, že problém s výkonem dané transakce je obecně známý

a již existuje doporučené řešení. V případě Z transakce je třeba případný špatně naprogramovaný kód řešit přímo s programátorem.[4]

Zvýšení průměrné doby odezvy mohou způsobit též samotní uživatelé svým nevhodným používáním systému. V mnoha transakcích uživatel volí rozsah dat, se kterými se bude daná operace provádět. Pokud bude rozsah dat značný, daná transakce se bude provádět dlouhou dobu bez ohledu na výkon nebo optimalizaci. Zmíněné transakce mohou skončit až po desítkách minut. Uživatel mezitím pokračuje v práci v jiných oknech aplikace, situace ho proto neomezuje. Nicméně velice výrazně stoupá vypočítaná průměrná doba odezvy systému. Správný postup je takový, že uživatel dlouho trvající transakce naplánuje jako úlohy na pozadí, a tak se již transakce do průměrného času odezvy pro dialog nepočítá. Uživatelům spouštění v dialogu vyhovuje více a změnit jejich návyky je úkol náročný.[4]

5.1.4 Časový profil

Zajímavou informací v transakci ST03 je též časový profil (Obr. 7). Tato tabulka ukazuje vytížení systému v jednotlivé hodiny a průměrnou dobu odezvy v těchto hodinách.

Interval	# Steps	T Response Time	Ø Time	Process.	Avg. Proc. Time	T CPU~	Ø CPU~
13--14	292	835	2.860,0	55.518	190,1	13	45,5
14--15	322	42	130,3	13.972	43,4	7	21,5
15--16	271	14	51,3	8.474	31,3	5	16,6
16--17	279	20	73,5	9.658	34,6	5	17,8
17--18	269	13	47,5	9.708	36,1	4	16,4
18--19	271	20	72,0	8.854	32,7	6	20,4
19--20	271	13	46,4	9.449	34,9	5	17,0
20--21	274	21	76,7	11.628	42,4	5	17,8
21--24	616	841	1.364,8	117.643	191,0	122	198,8

Obr. 7. ST03 – časový profil

Pokud výkonové problémy nastávají pouze v určitých časech, můžeme se v analýze na tuto dobu zaměřit. Zjistíme-li, že zvýšená doba odezvy je způsobena vyšší zátěží systému v tento čas, je třeba změnit některé naplánované úlohy tak, aby se spouštěly mimo tento problematický časový úsek.[4]

Kupříkladu on-line záloha databáze může poměrně zásadně ovlivnit celkový výkon systému. V situaci, kdy je zátěž systému rozdělena rovnoměrně, a nebo systém vykazuje stejnou dobu odezvy bez ohledu na změnu v zátěži, nepřinese nám časový profil pro naši analýzu žádnou zásadní informaci.

Je dobré se však stejným způsobem zaměřit na jednotlivé dny v týdnu. Měli bychom porovnávat zvlášť víkendové dny a zvlášť dny pracovního týdne. O víkendu systém využívá méně uživatelů. Většinou ale bývá naplánováno více úloh na pozadí a také některé úlohy údržby databáze. V závislosti na počtu uživatelů nebývá vyšší doba odezvy o víkendu vnímána negativně.

5.2 Nedostatečný hardware

V následujících odstavcích si osvětlíme některé ukazatele nedostatečně nadimenzovaného hardwaru. Problémy bývají nejčastěji s nedostačujícím výkonem procesorů, chybějící operační paměti a v menší míře pak propustností sítě či I/O diskovými operacemi.[4]

V případě procesorů (bývá jich vždy na serveru více) se uvádí jako maximální přípustná hranice 80% vytížení. Pokud je tedy průměrný „idle time“ menší jak 20%, je vhodné uvažovat o navýšení počtu procesorů. Zkontrolovat musíme též zaplnění odkládacího prostoru. Nemělo by překročit 20% hodnoty velikosti operační paměti. Pokud takovou situaci na serveru zjistíme, musíme se zaměřit, zda je tímto negativně ovlivněna průměrná doba odezvy. Pokud je processing time více než dvojnásobný oproti CPU time, značí to nedostatek procesorů na daném serveru.[4]

Pro kontrolu dostatku paměti musíme porovnat alokovanou virtuální paměť s velikostí fyzické paměti na serveru. Nemělo by být alokováno více jak 150% hodnoty fyzické paměti. Tato kontrola je závislá na použitém operačním systému.[4]

Příčiny problémů mohou být ve špatně rozložené zátěži. Tedy za předpokladu, že se náš systém SAP skládá z více instancí, které jsou fyzicky umístěné na různých serverech. Další příčinou mohou být procesy s velkou zátěží procesoru. Zde máme na mysli nejen work procesy systému SAP, ale též procesy databáze nebo i jiné procesy. Poslední možnou příčinou je nedostatečný hardware.

5.3 Nerovné rozložení zátěže

Pokud máme podezření na nerovnoměrné rozložení zátěže, je zapotřebí porovnat vytížení procesoru a obsazení paměti na různých serverech. Je též dobré porovnat průměrné doby odezvy jednotlivých SAP instancí podle příslušnosti k serverům. K zobrazení těchto informací použijeme v transakci ST03 přes výběr položek:

Load History and Distribution > Instance comparison > Day/Week/Month

5.4 Monitor operačního systému

Systém SAP může být provozován na různých operačních systémech (Windows, Linux, Unix). Zjišťování informací o vytížení hardwaru přímo v operačním systému by vyžadovalo rozličné znalosti v závislosti na variabilitě použitých řešení. Pro analýzu nám však výborně poslouží transakce monitoru operačního systému. Její kód je ST06 (Obr. 8). Případně můžeme použít cestu:

Architecture and Technology > System Administration > Performance > Operating system > Local > Operating System Monitor [4]

Local (ivml2005) / Operating System Monitor: Linux

Refresh display | Detail analysis menu | Operating System collector

Fri Apr 17 00:40:32 2015 interval 10 sec.

CPU

Utilization user %	7	Count	1
system %	3	Load average 1 min	0.13
idle %	64	5 min	1.85
io wait %	26	15 min	2.87
System calls/s	0	Context switches/s	1709
Interrupts/s	390		

Memory

Physical mem avail Kb	3653724	Physical mem free Kb	686808
Pages in/s	0	Kb paged in/s	0
Pages out/s	0	Kb paged out/s	0

Pool

Configured swap Kb	2658748	Maximum swap-space Kb	2658748
Free in swap-space Kb	2658748	Actual swap-space Kb	2658748

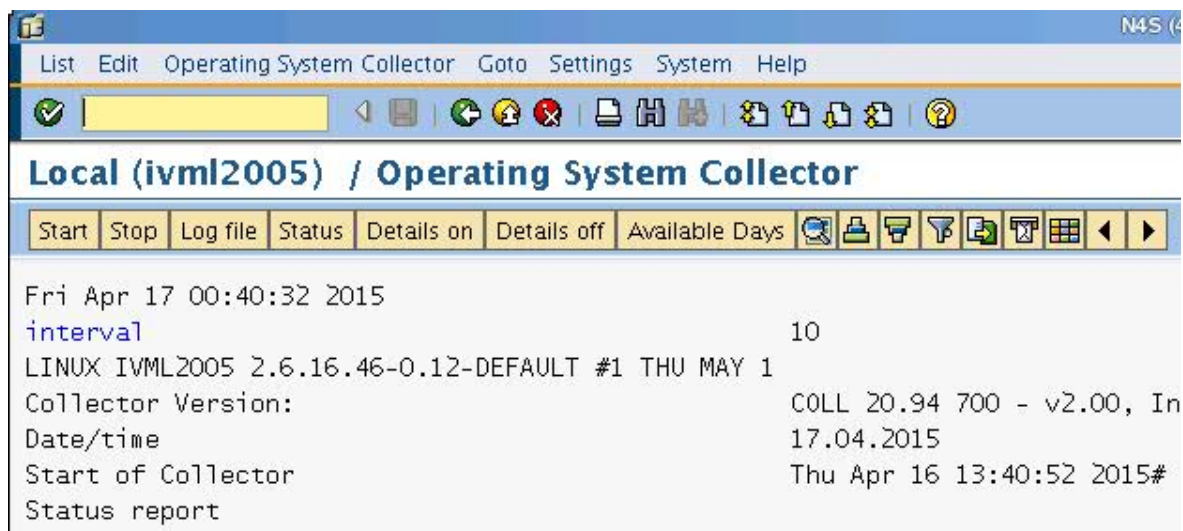
Obr. 8. Monitor operačního systému

Mějme na paměti, že se nám zobrazí hodnoty pro server, na kterém jsme aktuálně přihlášení. Pokud potřebujeme zjistit údaje o ostatních serverech, použijeme transakci OS07, kde zvolíme námi požadovaný server. Alternativně můžeme použít transakci SM51 a na daný server se přihlásit. Cesta pro výběr vzdáleného serveru je následující:

Architecture and Technology > System Administration > Performance > Operating system > Remote > Operating System Monitor [4]

Pro správnou funkčnost monitoru operačního systému je nezbytné, aby byl na daném serveru spuštěn proces saposcol (z anglického SAP operating system collector). Tento proces obstarává získávání údajů z operačního systému. Informace jsou aktualizovány každých 10 sekund. Jestliže se zobrazují v monitoru operačního systému (Obr. 9) nulové hodnoty nebo se neaktualizují, zaměříme se na funkčnost procesu saposcol. V transakci ST06 vybereme položku Operating System Collector a zkontrolujeme status kolektoru (tlačítko Sta-

tus). Dále můžeme překontrolovat log o činnosti kolektoru (tlačítko Log File) a podle potřeby kolektor restartovat (tlačítka Stop a Start) [1][4]



Obr. 9. Kolektor v ST06



Obr. 10. Proces kolektoru na OS

V monitoru operačního systému můžeme zjistit údaje o vytížení procesoru, zaplnění odkládacího prostoru, zaplnění operační paměti a též jejich historii. Standardně se nám zobrazují údaje za posledních 24 hodin. Starší údaje je možno zobrazit přes tlačítko Details.

5.5 Náročný proces

V praxi se často stává, že velká zátěž systému je způsobena pouze jedním procesem. Abychom takový proces identifikovali, použijeme opět transakci ST06, zvolíme Detail Analysis Menu a následně Top CPU. Zobrazí se nám tabulka s procesy, které nejvíce vytěžují procesor na daném serveru. Tabulka obsahuje tyto informace:

- ID procesu na operačním systému

- Vlastník procesu
- Jméno procesu
- Procentuální vytížení procesoru
- Celkový čas vytížení procesoru
- Alokovaná paměť
- Priorita [4]

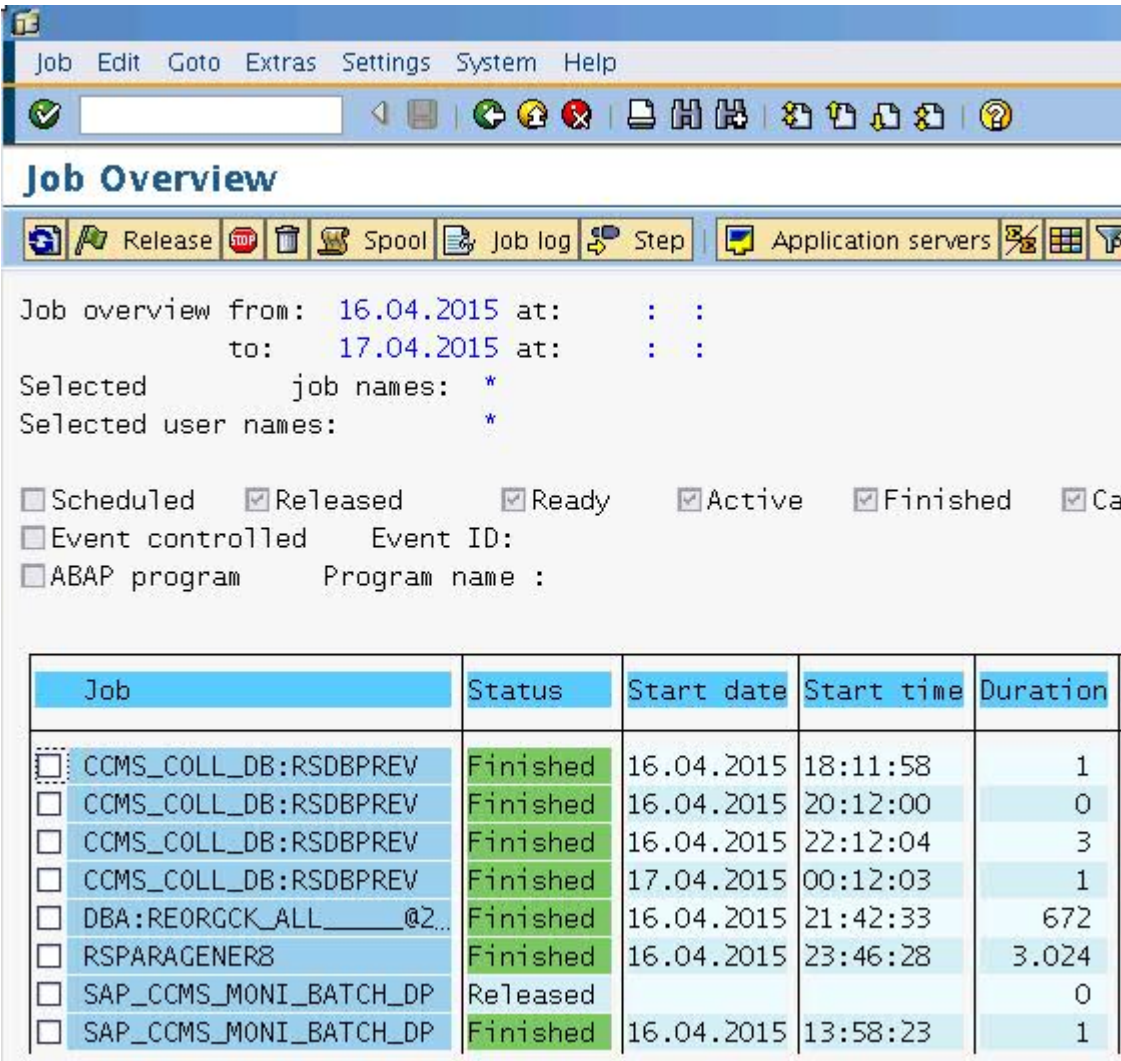
Tabulka je seřazena od nejnáročnějšího procesu k méně náročným. Jestliže zde nalezneme nějaký proces, který výrazně převyšuje v zatížení ostatní procesy, vyžaduje to bližší zkoumání. Pro představu uvádíme přehled možných procesů pro systém SAP využívající databázi Oracle a systém zálohování přes BR*Tools. Jedná se o častou konfiguraci.

Tab. 1. Procesy a jejich účel

<i>Jméno procesu</i>	<i>Účel</i>
disp+work	SAP work proces (Windows)
dw.<jméno_instance>	SAP work proces (Unix)
oracle<SID>	Proces databáze Oracle
ora_<cokoliv>_<SID>	Proces databáze Oracle
brbackup	Záloha databáze
brarchive	Záloha logů databáze
brspace	Úprava databáze (rozšíření, reorganizace)
brconnect	Kontrola databáze
backint	Spojení se serverem pro zálohy (např. TSM)
gw.<jméno_instance>	SAP Gateway
ms.<jméno_instance>	SAP Message server
icman	SAP Internet Communication Manager
saposcol	SAP OS Collector
<ostatní procesy>	Procesy operačního systému

Při zálohování databáze můžeme zaznamenat procesy s výrazně vyšší spotřebou zdrojů. Takové chování je naprosto v pořádku. Procesy operačního systému by se na celkové zátěži neměly podílet nikterak významně. Procesy systému SAP a databáze by měly spotřebovávat maximálně jednotky procent.

Zjistíme-li, že nejvýrazněji vytěžujícím procesem je některý proces systému SAP, můžeme snadno dohledat, který uživatel nebo úloha to způsobili. Poznamenejme si ID příslušného procesu a vyhledáme ho v transakci SM50, která zobrazuje přehled work procesů. Následně lze dále zjistit, zdali jsou nároky oprávněné. V případě dialogového work procesu můžeme kontaktovat příslušného uživatele, který dialog spustil. V případě úlohy na pozadí zjistíme detaily přes transakci SM37 (Obr. 11). Též máme možnost změnit čas startu úlohy tak, aby se spustila tehdy, kdy je systém méně vytížen. Náročnost úlohy pak nebude mít tak veliký vliv na práci uživatelů.[4]



Job Edit Goto Extras Settings System Help

Job Overview

Release Spool Job log Step Application servers

Job overview from: 16.04.2015 at: : :
to: 17.04.2015 at: : :
Selected job names: *
Selected user names: *

Scheduled Released Ready Active Finished Ca
 Event controlled Event ID:
 ABAP program Program name :

Job	Status	Start date	Start time	Duration
<input type="checkbox"/> CCMS_COLL_DB:RSDBPREV	Finished	16.04.2015	18:11:58	1
<input type="checkbox"/> CCMS_COLL_DB:RSDBPREV	Finished	16.04.2015	20:12:00	0
<input type="checkbox"/> CCMS_COLL_DB:RSDBPREV	Finished	16.04.2015	22:12:04	3
<input type="checkbox"/> CCMS_COLL_DB:RSDBPREV	Finished	17.04.2015	00:12:03	1
<input type="checkbox"/> DBA:REORGCK_ALL_@2...	Finished	16.04.2015	21:42:33	672
<input type="checkbox"/> RSPARAGENER8	Finished	16.04.2015	23:46:28	3.024
<input type="checkbox"/> SAP_CCMS_MONI_BATCH_DP	Released			0
<input type="checkbox"/> SAP_CCMS_MONI_BATCH_DP	Finished	16.04.2015	13:58:23	1

Obr. 11. Transakce SM37

Při zjištění, že nejvíce vytěžujícím procesem je program, který nenáleží systému SAP nebo databázi, musíme zjistit původ a funkci tohoto programu a následně rozhodnout, zdali tento

program může být vypnut či například přesunut na jiný server. Zde je řešení však velice individuální v závislosti na daném programu a nelze tedy doporučit nějaké obecné řešení.

5.6 Monitor disků

Monitor operačního systému nám poskytuje mnohem více informací, než jsme doposud uvedli. Další velice zajímavou funkcí je analýza čtení/zápisu pro jednotlivé disky. Informace si můžeme zobrazit přes Detail Analysis Menu, tlačítko Disk. Zobrazí se nám informace o discích za posledních 60 vteřin. Přehled popisků sloupců uvádíme v tabulce Tab. 2.[4]

Pokud nalezneme u některého z disků vytížení větší než 50%, je to důvod pro bližší pohled. Může to být způsobeno například skutečností, že se na disku nachází odkládací soubor operačního systému. Pro další analýzu využijeme možností, které nabízí operační systém či přímo výrobce disku.[4]

Tab. 2. Význam popisků

<i>Popisek</i>	<i>Význam</i>
Name	Jméno disku
Resp.	Průměrná doba odezvy v milisekundách
Util.	Procentuální vytížení
Queue Lng.	Počet čekajících procesů na čtení/zápis
Wait	Doba čekání v milisekundách
Serv.	Doba režie disku
TransfKB/s	Velikost přenesených dat
Operatn./s	Počet operací za sekundu

5.7 Monitor LAN

Výkonnostní problémy mohou být způsobeny vysokou dobou odezvy lokální sítě. Pro otestování volíme následující cestu:

ST06 > Detail Analysis Menu > LAN

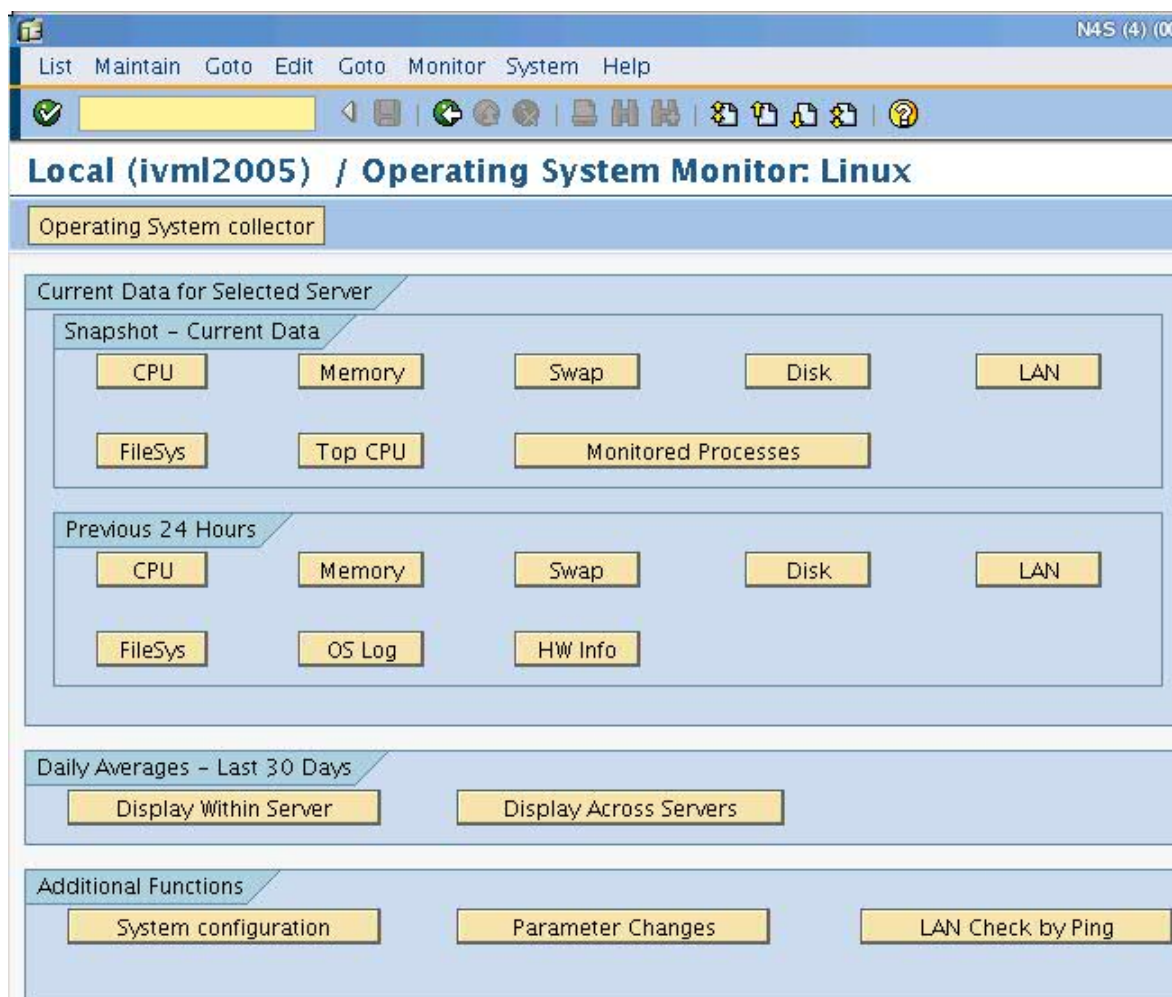
Můžeme vybrat databázový server, aplikační server nebo počítač s prezentační vrstvou a spustit jednou nebo vícekrát příkaz ping. Tak zjistíme hodnotu doby odezvy i případné ztráty dat při komunikaci po síti.[4]

5.8 Monitor změn parametrů

Pro operační systémy typu Unix jsou v OS monitoru zaznamenávány všechny změny parametrů. Pokud máme podezření, že za zhoršenou stabilitou či výkonností našeho SAP systému zodpovídá nějaká změna parametru na operačním systému, lze si historii změn zobrazit též v OS monitoru. Pro zobrazení použijeme následující cestu:

ST06 > Detail Analysis Menu > Parameter changes > History of file

Transakce ST06 poskytuje ještě další informace, které mohou být pro analýzu problému užitečné (Obr. 12).[4]



Obr. 12. Možnosti OS monitoru

6 VÝKON DATABÁZE

V současné době mohou systémy SAP používat mnoho různých databází. Nicméně určitý typ problému může nastat bez rozdílu použité databáze a postup k odstranění je na použité databázi též nezávislý. Pro získání informací o stavu databáze použijeme transakci Monitor výkonu databáze. Spouští se transakčním kódem ST04. Monitor získává informace prostřednictvím work procesů z databázového rozhraní a má přímo implementovány některé databázově závislé funkce. Pro spuštění transakce můžeme alternativně použít následující cestu:

Architecture and Technology > System Administration > Performance > Database > Activity [4]

6.1 Analýza bufferů databáze

Přestože v dalším textu je hojně užíván termín „buffer“, bude se někdy jednat o mezipaměť a bylo by proto vhodnější použít termín „cache“. Nicméně v systémech SAP i v odborné literatuře je vyhrazená paměť označována jako buffer, budeme se i my držet tohoto označení.[4]

Každá databáze má mnoho různých bufferů, které urychlují přístup k datům. Čtení dat z bufferu je řádově stokrát rychlejší než čtení z disku. Pokud jsou datové objemy příliš velké nebo buffery příliš malé, některá data jsou odstraněna z bufferu a při dalším čtení musejí být načítána z disku. To způsobuje značné zpomalování načítání dat. Proto je důležité buffery monitorovat.[1][4]

Nejdůležitější je buffer datový. Ten obsahuje části nejčastěji čtených databázových tabulek a jejich indexů. Datový buffer se skládá z jednotlivých bloků o velikosti 2 KB, 4 KB nebo 8 KB v závislosti na použité databázi a operačním systému.[4]

Klíčové jsou zde následující pojmy:

- Fyzické čtení (Physical read) – Počet stránek nebo bloků, které jsou načteny z disku.
- Logické čtení (Logical read) – Celkový počet stránek nebo bloků, na které bylo systémem přistoupeno. Jedná se o bloky načtené z bufferu i z disku.

- Kvalita bufferu (Buffer Quality) – Tato hodnota udává procentuální vyjádření, kolik požadavků bylo obslouženo přímo z bufferu, tedy podíl čtení z bufferu k počtu čtení celkem.

Čím vyšší číslo kvality bufferu, tím lépe. Maximální možná hodnota je 100 %. Po startu databáze jsou buffery prázdné, a proto několik prvních hodin bude kvalita bufferu nízká. Nevypovídá to však o špatně nastavené velikosti bufferu. Kvalitu bychom měli vždy ověřovat až po několika dnech provozu.[4]

Jedním z důvodů pro špatnou kvalitu bufferu mohou být i náročné SQL příkazy. Pokud je použit takový příkaz a databáze vrátí velký objem dat (např. celou tabulku), tak jsou z bufferu odstraněna původní data a jejich místo zaujme výsledek příkazu. Při dalším provozu se však opět načítají původní, často používaná data z disku, a dramaticky tím kvalita bufferu klesá.[4]

6.2 Exkluzivní zámky

Pokud je na určitém řádku tabulky prováděna změna (update), musí být tento řádek uzamčen exkluzivním databázovým zámkem. Pokud se další proces pokusí uzamknout stejný řádek, nastává čekání na exkluzivní zámek (exclusive lock wait). Pokud je doba čekání dlouhá, znamená to významné zhoršení výkonnosti databáze. Procesy musí čekat na uvolnění zámku a celá transakce se o tuto dobu prodlužuje. Pro zobrazení informací o čekání na exkluzivní zámek použijeme transakci DB01, nebo alternativně cestu:

Architecture and Technology > System Administration > Performance > Database > Exclusive locks

Transakce nám zobrazí následující údaje o zámcích:

- ID databázového procesu
- Server, ze kterého byl požadavek spuštěn
- Čas vzniku zámku a informaci o zámku

Pokud je zámek aktivní několik minut, měli bychom kontaktovat uživatele (vlastníka) a situaci s ním konzultovat. Může se jednat o špatně pracující program, který by mohl být ukončen. Další možností je násobné spuštění určitého programu, kdy si procesy zamkly zdroje navzájem a žádný z nich nemůže být dokončený. Zde je zapotřebí nastudovat dokumentaci k programu, či začít využívat program jiným způsobem.[4]

Při krátké době trvání zámků se jedná pravděpodobně o výkonnostní problém databáze. Příkazy jsou zpracovávány nedostatečně rychle, a proto další požadavky na zámek čekají ve frontě. V tomto případě jsou vzniklé zámky pouze následkem špatné výkonnosti databáze, nikoliv příčinou.

6.3 Záznam o stavu databáze

Každá databáze ukládá zprávy o své činnosti do určitého souboru. Tento soubor obsahuje důležité informace například o chybách, které během činnosti databáze nastaly. Tento soubor by měl být pravidelně kontrolován a lze jej prohlížet přímo ze systému SAP v následující cestě:

DB02 > Performance > Additional Functions > Alert Log [4]

Pokud máme podezření, že snížení výkonu je způsobené změnou parametru databáze, lze si všechny změny zkontrolovat v transakci DB02. K položkám aktivní parametry (Active parameters) a historie parametrů (Parameters history) se dostaneme pomocí následující cesty:

DB02 > Performance > Additional Functions > Database Parameters [4]

6.4 SQL aktivity transakce

Občas nastane situace, že si uživatel stěžuje na pomalé odezvy systému pouze v určité transakci, nebo při určité akci. Pokud nezjistíme žádné problémy v systému, lze ještě spustit sledování databázových příkazů (SQL) pro danou akci. Tato akce generuje poměrně rozsáhlý záznam o činnosti uživatele. Pro potřebu analýzy je tedy nezbytné maximálně omezit dobu, po kterou je sledování spuštěno.[4]

7 ANALÝZA KONFIGURACE PAMĚTI

Analýza paměťových bufferů je velice častý úkon administrátora systému při problémech s výkonností. Výborně nám poslouží transakce ST02, zobrazitelná též v následující cestě:

Architecture and technology > System administration > Monitor > Performance > Setup/Buffers > Buffers

Na úvod poznamenejme, že následující informace a postupy se vztahují vždy k určité instanci, na které jsme právě přihlášení. Pokud si přejeme instanci změnit, použijeme transakci SM51.[1] [4]

7.1 Monitor konfigurace paměti

Při zadání transakce ST02 se nám zobrazí přehled bufferů. Jedná se pouze o buffery, které přímo náleží systému SAP a dané instanci. Nenalezneme zde například buffery pro databázi nebo ostatní instance SAP. Jedná se vždy o údaj od posledního restartu dané instance. Datum je též uvedeno v přehledu. Význam jednotlivých sloupců přehledu vysvětluje následující tabulka:

Tab. 3. Sloupce přehledu bufferů

<i>Název sloupce</i>	<i>Význam</i>
Buffer	Jméno bufferu
HitRatio %	Pravděpodobnost, že se čtený údaj nachází v bufferu
Alloc. KB	Velikost bufferu
Freesp. KB	Volné místo v bufferu
% Free sp.	Procentuální vyjádření volného místa
Dir. Size	Maximální počet objektů v bufferu
FreeDirEnt	Počet volných míst pro objekty v bufferu
% Free Dir	Procentuální vyjádření volného počtu objektů
Swaps	Počet objektů odložených na disk
DB Accs	Počet přístupů do databáze pro daný buffer

Rozvedme si blíže některé výše zmíněné pojmy. Nejdůležitějším z nich je pravděpodobnost s jakou se čtený údaj nachází v bufferu. Je to obdoba vyjádření kvality bufferu, který se týkal bufferů databázových. Vyšší číslo znamená lepší hodnotu. Maximálně lze dosáhnout 100 % a to v případě, že jsou všechny požadavky zodpovězeny přímo z bufferu.

Swap určuje počet objektů, které byly odloženy na disk. Taková situace může nastat, pokud nedostačuje velikost vyhrazené paměti, a nebo je zvolen nízký počet objektů pro buffer.[4]

Na první pohled je zřejmé, které buffery odkládají objekty na disk. V přehledu je nenulový počet zvýrazněn červenou barvou. Ne vždy to však znamená problém. Pokud je počet nízký v porovnání s celkovým počtem čtení, pak se nejedná v zásadě o problém. Směrodatná je následně zejména hodnota HitRatio.[1] [4]

Důležitým pojmem je zneplatnění objektu. Děje se tak při nahrání nové verze programu, nebo úpravě objektů. Například v důsledku transportu změn ze systému kvality nebo vývoje. V bufferu se pak takový objekt zneplatní a musí být znovu načten. To má za následek snížení hodnoty HitRatio.

Situaci vyhodnocujeme jako problém, pokud nastanou obě následující události současně:

- Hodnota HitRatio je nízká
- Počet odložených objektů je vysoký

Při splnění výše zmíněných podmínek můžeme přistoupit k zvýšení hodnoty bufferu. Odkládání na disk může být způsobeno chybějícím volným místem v bufferu, a nebo nízkou hodnotou počtu objektů. V případě nízkého počtu objektů se nebojme tuto hodnotu zdvojnásobit. Pokud je problémem nedostatek volného místa v bufferu, můžeme tento buffer zvětšit za předpokladu, že na serveru je dostatek volné paměti. Buffer obvykle zvětšujeme o 10% - 50% stávající hodnoty.[4]

Tune Edit Goto Environment Monitor System Help

Tune: Detail Analysis (ivml2005_N4S_42)

Current parameters Buffered objects

System: ivml2005_N4S_42
Date + Time of Snapshot: 17.04.2015 11:27:18

Efficiency	HITRATIO	%	34
	HITS		112
	REQUESTS		326
	DB access quality	%	100
	DB access		0
	DB access saved		0
	Reorgs		0
Size	Allocated	KB	4.096
	Available	KB	3.384
	Used	KB	130
	Free	KB	3.243
Directory entries	Available		2.000
	Used		62
	Free		1.938
Swaps	Objects swapped		0
	Frames swapped		0
Resets	Total		0

Obr. 13. ST02 – detail bufferu

Přestože hodnota HitRatio je pouze 34%, nejedná se v zásadě o problém. Nepochází zde k odkládání na disk. Nízká hodnota je způsobena krátkou dobou běhu systému.

8 SAP HANA

SAP HANA je databázový systém společnosti SAP SE, který byl vyvinut především pro potřeby systémů SAP. Systém vznikl na základě již existujícího softwaru společnosti SAP SE, a to:

- TREX – software pro urychlení vyhledávání v databázi
- MaxDB – databáze
- LiveCache – systém s daty dostupnými zvláště v operační paměti
- P*TIME – databáze

Databáze používá pro dotazy jazyk SQL a pro uživatelské rozhraní kód HTML5.[6]

8.1 Porovnání s konvenčními databázemi

Systém SAP může využít různé databáze. Podle našich zkušeností je nejčastěji použita databáze Oracle. Na výběr však máme ještě databáze MSSQL, DB2, MaxDB a v neposlední řadě samozřejmě SAP HANA. Posledně jmenovaná databáze dokáže poskytovat data výrazně rychleji než ostatní „konvenční“ databáze. Výrazného navýšení rychlosti je dosaženo mnoha inovacemi.

Databáze SAP HANA je nazývána In memory database. Značí to skutečnost, že data jsou dostupná v operační paměti serveru. Díky tomu lze dosahovat opravdu velkého zrychlení oproti čtení dat z disku. To ale není jedinou inovací oproti ostatním typům databází. V databázi SAP HANA se data nacházejí v komprimovaném formátu. To klade nižší nároky na propustnost i velikost paměti. Kompresní poměr je přibližně 6:1.[6]

U konvenčních databází platí, že databázová vrstva poskytuje data aplikační vrstvě a ta následně s daty provádí další operace. V případě databáze SAP HANA je snaha taková, aby co nejvíce výpočtů a analýz proběhlo již na databázové vrstvě a aplikační vrstva obdržela pouze výsledky. Minimalizuje se tím přenos dat mezi oběma vrstvami. Tímto se celá operace urychlí.[6]

8.2 Ukládání dat

Nejprve osvětleme rozdíl mezi databázemi typu OLAP a OLTP. Jestliže je náš systém zaměřen na výpočty či analýzy (OLAP), budou často prováděny matematické operace nad

jednotlivými sloupci tabulek. Například výpočet průměru či celkový součet. V tomto případě bude v databázi načítán vždy určitý sloupec tabulky, a je tedy velice výhodné mít tato data uložena v jednom bloku.

Pro databáze typu OLTP jsou data ukládána po řádcích. To přináší výhodu při vkládání, mazání či změně řádku. Data jsou načtena v jednom celku, zatímco u OLAP bychom načítali z více bloků, protože každá položka sloupce je uložena v jiném bloku dat.[6]

Podle typu systému je tedy (konvenční) databáze zvolena OLAP či OLTP v závislosti na hlavní funkci systému. Zde přináší SAP HANA další inovaci, a to kombinaci obou technologií. Podle funkce jsou některé tabulky uloženy po řádcích a jiné po sloupcích. Takto se dosáhne optimálního uložení, a práce s databází je rychlejší.[6]

8.3 Hardware pro SAP HANA

Pro databázi SAP HANA nemůžeme použít libovolný hardware nebo operační systém. Databáze je prodávána jako celé „SAP HANA Appliance“ řešení, které obsahuje potřebný hardware, na kterém je již nainstalován operační systém a databáze SAP HANA. Tím je zaručeno, že všechny komponenty budou optimálně spolupracovat. Výběr výrobců hardwaru pro SAP HANA Appliance je omezený. Certifikované jsou pouze firmy IBM, Hewlett Packard, Fujitsu Computers, CISCO systems a DELL.[6]

Data v databázi jsou dostupná v operační paměti, a proto servery pro SAP HANA obsahují až jednotky TB RAM. Vždy použita Intel X86 architektura.[6]

8.4 Přejít na SAP HANA

Jak již bylo zmíněno, databáze SAP HANA dokáže poskytovat data mnohem rychleji než konvenční databáze. Implementuje mnoho nových a převratných postupů a technologií. Mohli bychom nabýt dojmu, že přechod na tuto databázi je jasnou volbou. Uvedme však též argumenty proti přechodu na SAP HANA. Prvním problémem je nemožnost využít náš stávající hardware, na kterém provozujeme naši konvenční databázi. Při přechodu bychom tedy znovu investovali do nového hardwaru (jako součást SAP HANA Appliance).

Dále musíme posoudit, jaké zrychlení by nám přechod přinesl. Velká výhoda dostupnosti dat v paměti se projeví pouze při čtení dat. Avšak při zápisu dat již výhoda nenastává. Data se musí zapsat na úložiště (disky) stejně jako u konvenční databáze, aby při případném výpadku napájení nedošlo ke ztrátě dat. Z toho důvodu je tedy databáze SAP HANA

vhodná zejména pro systémy typu OLAP, kde se provádí převážně analýzy (čtení) a v menší míře pak databázové transakce (zápis).[6]

Pokud u konvenční databáze přestane dostačovat volné místo na disku pro data, celkem snadno můžeme dokoupit další disk a připojit ho do systému. U databáze SAP HANA toto neplatí. Nemůžeme sami konfigurovat hardware a datová kapacita disků vždy koresponduje s celkovým množstvím paměti RAM tak, aby všechna data mohla být do této paměti nahrána. V případě požadavku rozšíření budeme nuceni koupit celý nový výkonnější server. Alternativní možností je koupit stejný server a připojit ho do clusteru. V obou případech je to poměrně nepružné řešení. Nelze navýšit kapacitu třeba o 20%.[6]

Závěrem této kapitoly lze doporučit, že přechod na databázi SAP HANA musí být pečlivě zvážen zvláště po finanční stránce. Základem je dobře pochopit business procesy a odhadnout dopady implementace SAP HANA.

II. PRAKTICKÁ ČÁST

9 OPTIMALIZAČNÍ PROJEKT

V následujících kapitolách budou představeny praktické ukázky z optimalizační činnosti. Pro čtenáře budou výbornou pomocí a vodítkem pro realizaci vlastního optimalizačního projektu. Připomeňme, že optimalizace zahrnuje celou škálu akcí. V první řadě je to pochopení požadavků zadavatele či porozumění nárokům společnosti na cílový systém.

9.1 Klíčové ukazatele výkonnosti

Častokrát zahajujeme optimalizaci z podnětu uživatelů, kteří si stěžují na špatný výkon systému. Taková definice problému je poměrně neurčitá, stejně jako požadavek uživatele, aby „systém běžel rychleji“. Proto je třeba si stanovit měřitelné klíčové ukazatele výkonnosti. Je to objektivní způsob, jak změřit výkonnost systému a prokázat zlepšení jako důsledek naší optimalizace. V anglické literatuře se tyto ukazatele vyskytují pod zkratkou KPI, která pochází ze slov Key Performance Indicator. Záleží vždy na konkrétní situaci a problému uživatele. Nicméně lze zmínit některé obecné ukazatele výkonnosti jako jsou například:

- Průměrná doba odezvy dialogu
- Průměrná doba odezvy ostatních akcí
- Kvalita bufferů
- Doba běhu vybraných úloh na pozadí
- Průměrné zpoždění startu úloh
- Spokojenost uživatelů se systémem

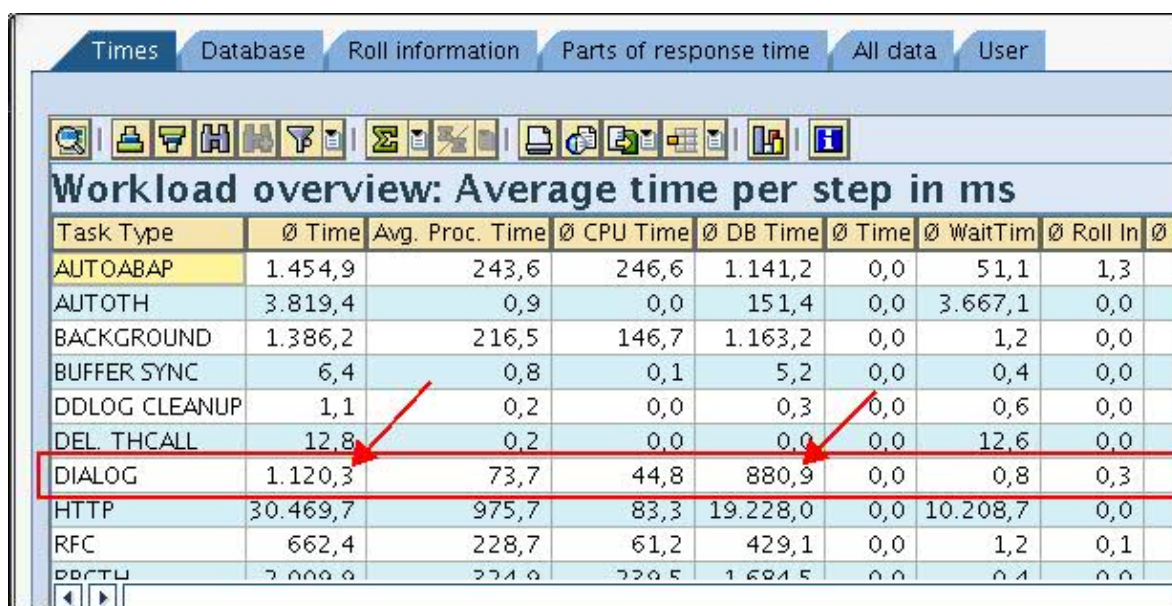
9.2 Zlepšení průměrné doby odezvy

Vezměme si tedy častý případ, že si uživatel stěžuje na výkon systému. S uživatelem je třeba diskutovat následující:

- Týká se špatný výkon dialogových kroků, úloh na pozadí, nebo jiné části práce?
- Týká se špatný výkon všech transakcí/úloh, nebo jen některých?
- Pociťují špatný výkon všichni uživatelé?

V našem příkladu uvažujme, že uživatel vnímá výkonnostní problém při spouštění transakcí (tedy dialogových kroků), problém se týká všech spouštěných transakcí a ostatní uživatelé mají podobné zkušenosti.

Na základě rozhovoru s uživatelem můžeme tedy vyloučit příčiny jako jsou neoptimalizovaný programový kód nebo pomalé síťové připojení uživatele k systému. Problém bude spíše ve špatné výkonnosti systému obecně. Klíčový ukazatel výkonnosti bude v tomto případě průměrná doba odezvy dialogového kroku. První kroky analýzy povedou do transakce ST03 ke zjištění aktuálního stavu.



Task Type	Ø Time	Avg. Proc. Time	Ø CPU Time	Ø DB Time	Ø Time	Ø WaitTim	Ø Roll In	Ø
AUTOABAP	1.454,9	243,6	246,6	1.141,2	0,0	51,1	1,3	
AUTOTH	3.819,4	0,9	0,0	151,4	0,0	3.667,1	0,0	
BACKGROUND	1.386,2	216,5	146,7	1.163,2	0,0	1,2	0,0	
BUFFER SYNC	6,4	0,8	0,1	5,2	0,0	0,4	0,0	
DDLOG CLEANUP	1,1	0,2	0,0	0,3	0,0	0,6	0,0	
DEL. THCALL	12,8	0,2	0,0	0,0	0,0	12,6	0,0	
DIALOG	1.120,3	73,7	44,8	880,9	0,0	0,8	0,3	
HTTP	30.469,7	975,7	83,3	19.228,0	0,0	10.208,7	0,0	
RFC	662,4	228,7	61,2	429,1	0,0	1,2	0,1	
PRCTU	2.000,0	224,0	220,5	1.684,5	0,0	0,4	0,0	

Obr. 14. ST03 – Workload overview

Z přehledu je patrné, že průměrný čas odezvy je 1.12s, což je poměrně rychlá odezva. Jedná se o čas od odeslání požadavku do jeho vyřízení. Nicméně bychom měli zaznamenat též informaci, že z celkového času odezvy připadá poměrně velká část na databázi. Zatím se tím neznepokojujme.

Je nepravděpodobné, že by si uživatel stěžoval na nízký výkon systému při času odpovědi 1.12s. Je však důležité zmínit, že se jedná o průměrný čas odpovědi, a může se výrazně měnit během dne. Pokračujme dále v naší analýze zobrazením časového profilu.

Interval	# Steps	T Re...	Ø Time	Process.	Avg. Pro...	T CPU~	Ø CPU~	T DB...	Ø DB Ti...
13--14	292	835	2.860,0	55.518	190,1	13	45,5	683	2.337,9
14--15	322	42	130,3	13.972	43,4	7	21,5	23	72,1
15--16	271	14	51,3	8.474	31,3	5	16,6	5	17,1
16--17	279	20	73,5	9.658	34,6	5	17,8	10	35,3
17--18	269	13	47,5	9.708	36,1	4	16,4	2	8,5
18--19	271	20	72,0	8.854	32,7	6	20,4	10	36,5
19--20	271	13	46,4	9.449	34,9	5	17,0	2	8,9
20--21	274	21	76,7	11.628	42,4	5	17,8	9	31,6
21--24	987	985	998,2	139.787	141,6	132	133,7	798	808,8

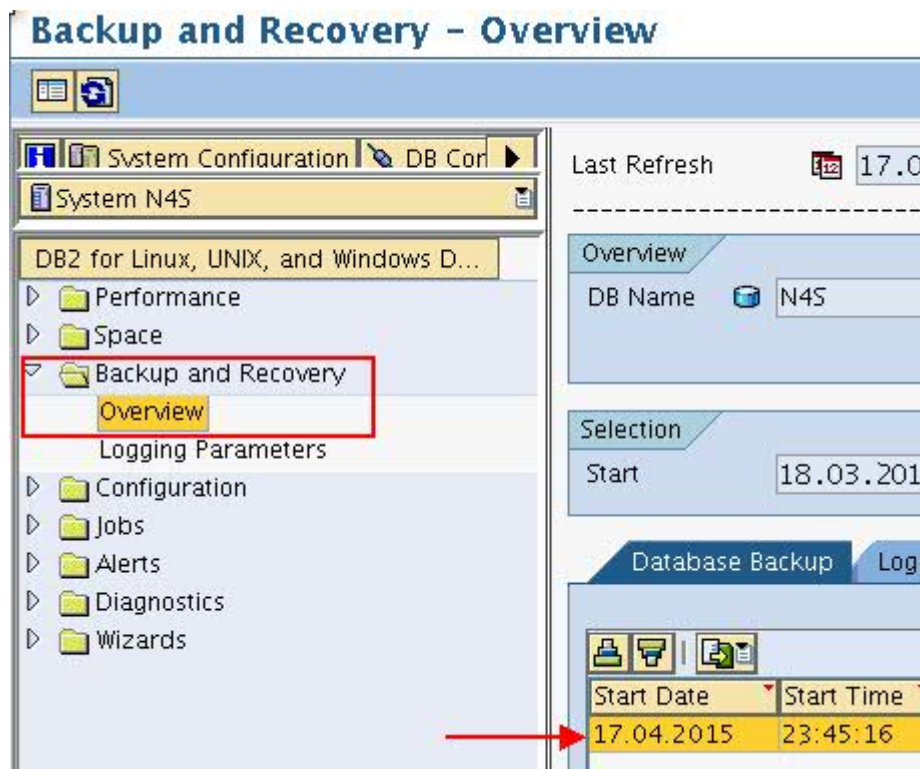
Obr. 15. ST02 – časový profil

A nyní již je patrný důvod uživatelské stížnosti. Mezi 13. a 14. hodinou došlo k výraznému nárůstu průměrného času odezvy, a to téměř na 3 vteřiny. To už je vnímáno jako špatná výkonnost systému. Též si všimněme, že majoritní část obsazuje čas načítání z databáze. Zatímco ostatní hodiny jsou časy zpracování požadavku a načítání z databáze podobné, zde je čas pro databázi více než desetinásobný. To by nemělo nastat.

Shrňme si doposud zjištěné informace:

- Problém se vyskytl pouze mezi 13. a 14. hodinou
- Problém je způsoben dlouhotrvajícím načítáním dat z databáze
- Problém není způsoben vyšším počtem kroků v daný čas (Steps)

Vzhledem k výše uvedenému bude problém v jednorázové akci, která přespříliš vytíží databázi a ostatní požadavky se proto vyřizují mnohem delší dobu, než je obvyklé. První nás jistě napadne, že v inkriminované době probíhala záloha databáze. Při online záloze je systém pro práci běžně dostupný, dochází však k načítání celého obsahu databáze, a to může mít negativní vliv na vyřizování ostatních požadavků, zvláště při větším zatížení systému. Zkontrolujme tedy čas zálohy databáze přes transakční kód DB02.



Obr. 16. DB02 – čas zálohy databáze

Z obrázku (Obr. 16) vyplývá, že náš problém záloha databáze nezpůsobila, protože začátek zálohování je nastaven na čas 23:45. Další pravděpodobnou variantou je náročná úloha na pozadí. Zkontrolujeme přehled úloh v transakci SM37 a zaměříme se na dotčené časové rozmezí.

Job	Status	Start time	Duration
<input type="checkbox"/> SAP_CCMS_MONI_BATCH_STARTUP_DP	Finished	13:00:49	52
<input type="checkbox"/> CCMS_COLL_DB:RSDB_PAR	Finished	13:00:50	1
<input type="checkbox"/> SAP_CCMS_MONI_BATCH_DP	Finished	13:11:47	1
<input type="checkbox"/> SAP_COLLECTOR_FOR_NONE_R3_STAT	Finished	13:11:47	1
<input type="checkbox"/> SAP_COLLECTOR_FOR_PERFMONITOR	Finished	13:11:47	4
<input type="checkbox"/> CCMS_COLL_DB:RSDB_PAR	Finished	13:11:48	0
<input type="checkbox"/> RSPARAGENS	Finished	13:16:26	2.742
<input type="checkbox"/> SAP_RSICFDLT	Finished	13:30:48	0
<input type="checkbox"/> SAP_RSICFDLT	Finished	13:41:47	1
<input type="checkbox"/> SAP_CCMS_MONI_BATCH_DP	Finished	13:53:48	5
<input type="checkbox"/> SAP_COLLECTOR_FOR_NONE_R3_STAT	Finished	13:53:56	3
<input type="checkbox"/> SAP_CCMS_MONI_BATCH_DP	Finished	14:11:47	9

Obr. 17. SM37 – přehled úloh

Při pohledu na přehled úloh (Obr. 17) zjišťujeme, že jedna úloha trvala výrazně delší dobu než ostatní. Zjistíme detaily o této úloze od vlastníka úlohy, nebo vyhledáním na Internetu. V našem konkrétním případě se jedná o program RSPARAGENER8, který kompiluje programový kód. Spouští se obvykle po změnách programového kódu jako je přechod na novější verzi komponenty, či po úpravách kódu samotného. Pokud se tato procedura po změnách neprovede, kód je kompilován vždy při prvním použití. To způsobí výrazné navýšení času pro uživatele, který daný kód (transakci, report) spustí jako první. Ideálním řešením je program RSPARAGENER8 spustit v době, kdy uživatelé se systémem nepracují.

Výsledkem naší druhé analýzy bude doporučení administrátorovi/uživateli, aby spuštění programu RSPARAGENER plánoval v nočních hodinách. Příslušná transakce pro tuto hromadnou kompilaci je SGEN.

9.3 Změna paměťových parametrů

V další analýze se zaměříme na buffery systému SAP. Kontrola by se měla provádět pravidelně a při zjištění problémů by se měla naplánovat náprava. Klíčový ukazatel výkonnosti v tomto případě bude kvalita bufferu a počet odkládání objektů na disk. Spustíme transakci ST02 a zkontrolujeme stav.

Buffer	HitRatio %	Alloc. KB	Freesp. KB	% Free Sp.	Dir. Size	FreeDirEnt	% Free Dir	Swaps
program	71,24	150.000	122	0,08	37.500	29.318	78,18	19.955
CUA	99,19	3.000	1.858	75,38	1.500	1.466	97,73	0
Screen	99,49	4.297	3.887	94,97	2.000	1.958	97,90	0
Calendar	100,00	488	360	76,43	200	50	25,00	0
OTR	100,00	4.096	3.384	100,00	2.000	2.000	100,00	0
Tables								0
Generic Key	97,78	29.297	25.362	90,87	5.000	250	5,00	44
Single record	62,49	10.000	5.407	54,88	500	455	91,00	1
Export/import	30,44	4.096	3.268	96,57	2.000	1.942	97,10	0
Exp./ Imp. SHM	100,00	4.096	3.384	100,00	2.000	2.000	100,00	0

Obr. 18. ST02 – přehled bufferů

Z obrázku je patrné, že tři buffery využívají swap. Zhodnotíme situaci:

- Program buffer – nízká kvalita, vysoký počet swapů => problém
- Generic Key buffer – vysoká kvalita, nízký počet swapů => OK

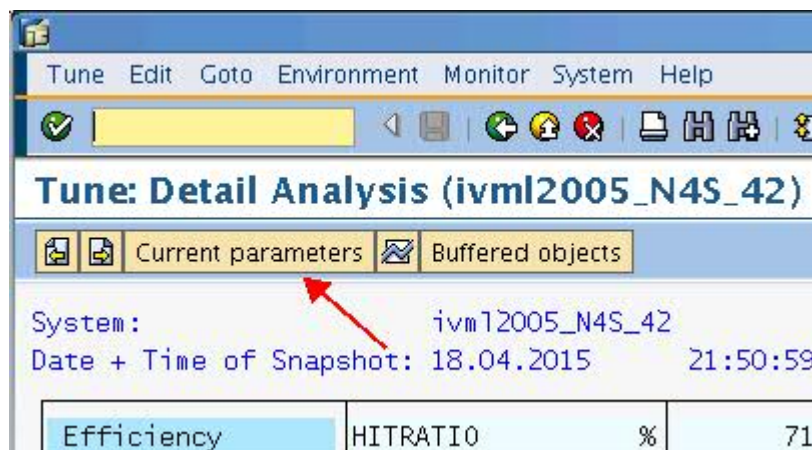
- Single record buffer – nízká kvalita, nízký počet swapů => OK

Bliže se tedy podíváme na program buffer, který je v obrázku označen. Dvojklikem přejdeme na obrazovku s více detaily (Obr. 19).

Efficiency	HITRATIO	%	71
	HITS		1.060.968
	REQUESTS		1.489.266
	DB access quality %		95
	DB access		84.525
	DB access saved		1.615.964
	Reorgs		0
Size	Allocated	KB	150.000
	Available	KB	144.836
	Used	KB	144.714
	Free	KB	122
	Caps	KB	122
Directory entries	Available		37.500
	Used		8.182
	Free		29.318
Swaps	Objects swapped		19.955
	Frames swapped		0
Resets	Total		0

Obr. 19. Detail program bufferu

Kvalita, respektive tedy úspěšnost, že se námi požadovaný údaj nachází v bufferu, je pouze 71%, což není dobrá hodnota. Zkontrolujeme, zdali schází volné místo v bufferu, nebo byl podhodnocen maximální počet objektů. V našem případě schází volné místo. Budeme tedy navyšovat příslušný paměťový parametr. Informace o souvisejícím parametru či parametrech získáme prostřednictvím tlačítka Current parameters:



Obr. 20. Volba pro zobrazení parametrů

A nyní se dostáváme k příslušným parametrům:

Buffer Name	Comment		
Prof.Param	Value	Unit	Comment
Program buffer PXA			
abap/buffersize	150000	kB	Size of program buffer
abap/pxa	shared		Program buffer mode

Obr. 21. Příslušné parametry

K našemu program bufferu se vztahují dva parametry (Tab. 5).

Tab. 4. Příslušné parametry

Parametr	Hodnota	Význam	Doporučení
abap/pxa	shared	mód bufferu	nelze změnit
abap/buffersize	150000	velikost bufferu	zvětšit hodnotu

Pro zvětšení bufferu budeme navyšovat hodnotu parametru abap/buffersize. Nejprve se na parametr podíváme blíže přes transakci RZ11 (Obr. 22)

Display Profile Parameter Attributes

Documentation

Param. Name
abap/buffersize

Short description(Engl) Program Buffer Size

Appl. area ABAP

ParameterTyp Integer value

Changes allowed Change permitted

Valid for oper. system All operating systems

Minimum

Maximum

DynamicallySwitchable

Same on all servers

Dflt value 150000

ProfileVal 150000

Current value 150000

Obr. 22. RZ11 – parametr abap/buffersize

Každý parametr má svoji základní hodnotu. V případě, že se daný parametr nachází v profilu instance s jinou hodnotou, použije se tato hodnota z profilu. A konečně, pokud je parametr dynamicky měnitelný (tzn. za běhu systému), může být aktuální hodnota ještě jiná.

Pro tento konkrétní případ zvolíme dvojnásobné navýšení. Jedná se vždy o určitý odhad na základě zkušeností a zjištěných statistik bufferu. Zde není možné parametr dynamicky měnit, provedeme tedy změnu v transakci RZ10 (Obr. 23).

Edit Profiles

Create Check Copy Import

Profile N4S_DVEBMGS42_N4SHOST

Version 000007

Obr. 23. Změna profilu

Maintain Profile 'N4S_DVEBMGS42_N4SHOST' Version '000006'

Copy Parameter Parameter Parameter

19.04.2015 Active parameters

Parameter Name	Parameter value
abap/buffersize	300000
sapgui/user_scripting	TRUE
rdisp/max_wprun_time	900
SAPSYSTEMNAME	N4S
SAPSYSTEM	42
INSTANCE_NAME	DVEBMGS42

Obr. 24. Změna parametru

Následně musíme restartovat systém, protože parametr není dynamicky měnitelný. Tento restart nám také vyprázdní obsah bufferů a též související statistiky. Musíme tedy počkat, než se buffery zaplní a statistiky budou vypovídající. Následně provedeme kontrolu, zdali došlo ke zlepšení situace (Obr. 25). Kvalita bufferu je nyní v pořádku.

Efficiency	HITRATIO	%	100
	HITS		870.306
	REQUESTS		873.049
	DB access quality	%	99
	DB access		8.169
	DB access saved		981.289
	Reorgs		0

Obr. 25. Vyhodnocení změny

9.4 Optimální rozložení úloh

Uživatelé si mohou stěžovat, že jejich úlohy běží déle, než je obvyklé. Ukážeme si, jak situaci řešit a problém odhalit. Klíčový ukazatel výkonnosti bude celková doba běhu úlohy na pozadí. Analýzu bychom mohli začít stejným způsobem jako v případě zvýšené doby odezvy pro dialogové kroky. Zaměřili bychom se na položku „BACKGROUND“, která odpovídá úlohám na pozadí. Obrázky postupu jsou uvedeny u příslušné kapitoly výše. V našem současném případě však touto cestou neodhalíme žádné nesrovnalosti. Pokračujeme tedy kontrolou přehledu úloh samotných.

Job	Status	Duration	Delay (sec.)
<input type="checkbox"/> SAP_COLLECTOR_FOR_NONE_R3_STAT	Finished	28	1.131
<input type="checkbox"/> SAP_COLLECTOR_FOR_PERFMONITOR	Finished	60	1.130
<input type="checkbox"/> SAP_CCMS_MONI_BATCH_DP	Finished	14	710
<input type="checkbox"/> SAP_COLLECTOR_FOR_NONE_R3_STAT	Finished	4	710
<input type="checkbox"/> SAP_COLLECTOR_FOR_PERFMONITOR	Finished	2	710
<input type="checkbox"/> SAP_RSICFDLT	Finished	8	526
<input type="checkbox"/> SAP_CCMS_MONI_BATCH_STARTUP_DP	Finished	282	395
<input type="checkbox"/> SAP_CCMS_MONI_BATCH_STARTUP_DP	Finished	318	384
<input type="checkbox"/> CCMS_COLL_DB:RSDBPREV	Finished	32	210
<input type="checkbox"/> SAP_CCMS_MONI_BATCH_DP	Finished	3	133
<input type="checkbox"/> SAP_RSICFDLT	Finished	2	126
<input type="checkbox"/> SAP_COLLECTOR_FOR_NONE_R3_STAT	Finished	10	123

Obr. 26. SM37 – přehled úloh

Všimněme si velkého zpoždění startu úloh. Pro lepší názornost byl výpis ještě seřazen a některé sloupce eliminovány. Zpoždění startu je způsobeno čekáním na zdroje. Ve většině případů je tím chybějícím zdrojem volný work proces. Podíváme se tedy na přehled work procesů na systému – transakce SM50 (Obr. 27).

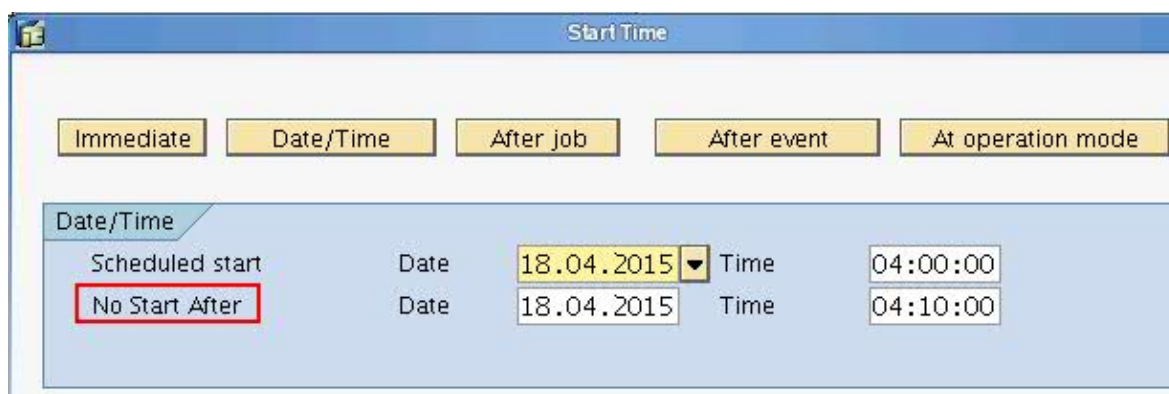
No.	Type	PID	Status	Runtime	Report	Cl.	User Names
0	DIA	6692	Running	4	SAPMSSY6	000	SAPSYS
1	DIA	6693	Running		SAPLTHFB	001	SAP*
2	DIA	6694	Running		SAPMSSY2	000	SAPSYS
3	DIA	6695	Waiting				
4	DIA	6696	Waiting				
5	DIA	6697	Waiting				
6	UPD	6698	Waiting				
7	ENQ	6699	Waiting				
8	BGD	6700	Running	329	CL_SQL_STATEMENT...	001	SAP*
9	BGD	6701	Running	105	CL_SQL_STATEMENT...	001	SAP*
10	BGD	6702	Running	54	CL_SQL_STATEMENT...	001	SAP*
11	BGD	6703	Running	12	CL_SQL_STATEMENT...	001	SAP*
12	SPO	6704	Waiting				
13	UP2	6705	Waiting				

Obr. 27. SM50 – přehled work procesů

Nyní je příčina zřejmá. Naše instance má 4 work procesy pro obsluhu úloh (typ BGD). Všechny jsou obsazené, což značí status „Running“. Každá další úloha musí čekat na volný work proces. Občas tento jev v systému nastává, aniž by představoval problém. Nesmí to být však trvalý stav.

Všichni uživatelé při plánování úloh obvykle používají začátky hodin, jako například 13:00, a nevyužívají pro start celý průběh určité hodiny (13:09, 13:26, 13:41). To má za následek velký počet úloh naplánovaných na začátek každé hodiny. Systém nedisponuje odpovídajícím množstvím work procesů, a některé úlohy tak musí čekat. Nicméně toto čekání nesmí nastat u všech úloh a nesmí přesáhnout určitou přijatelnou hodnotu.

Pokud se úloha spouští pravidelně, mívá obvykle nastavený čas, po kterém se již nespustí (Obr. 28). V případě neustálého zpoždování všech úloh může nastat situace, že se některé periodické úlohy nespustí nikdy.



Obr. 28. SM37 – čas spuštění

Vraťme se k našemu případu plně obsazených BGD work procesů. Mnohem více znepokojující je fakt, že je obsazena i velká část work procesů DIA. Pakliže není volný work proces typu DIA, nelze se na systém přihlásit a též nelze provádět žádné dialogové transakce. To už je vnímáno uživateli jako nedostupnost systému.

Navrhněme tedy takové řešení, kdy navýšíme počet dialogových work procesů a zároveň nastavíme operační módy tak, aby se nově přidávané dialogové work procesy změnil na typ BGD v nočních hodinách. Tím zajistíme dostatečný počet work procesů pro naše úlohy a nebude již docházet ke zpoždění. Počet dialogových work procesů zvýšíme prostřednictvím parametru `rdisp/wp_no_dia`. Změna parametru je popsána v předchozí kapitole, nebu-

deme se tím nyní blíže zabývat. Následně je třeba systém restartovat a poté můžeme již zkontrolovat navýšení počtu dialogových work procesů z transakce SM50:

Process Overview

No.	Type	PID	Status	Runtime	Report	Cl.	User
0	DIA	2966	Waiting				
1	DIA	2967	Waiting				
2	DIA	2972	Waiting				
3	DIA	2973	Running		SAPLTHFB	001	SAP*
4	DIA	2974	Waiting				
5	DIA	2976	Waiting				
6	DIA	2977	Waiting				
7	DIA	2978	Waiting				
8	DIA	2979	Waiting				
9	DIA	2980	Waiting				
10	UPD	2981	Waiting				
11	ENQ	2982	Waiting				
12	BGD	2998	Waiting				
13	BGD	3000	Waiting				
14	BGD	3001	Waiting				
15	BGD	3002	Waiting				
16	SPO	3003	Waiting				
17	UP2	3111	Waiting				

Obr. 29. SM50 – navýšený počet work procesů

Počet dialogových work procesů je tedy zvýšen na 10. Nyní nastavme operační módy pro změnu na typ BGD v nočních hodinách. Uvažujme, že uživatelé našeho systému SAP začínají pracovat nejdříve v šest hodin ráno a končí nejpozději v deset hodin večer. Tento časový interval ponechme work procesy beze změny. Mimo tento čas však nastavme změnu čtyřech dialogových work procesů na čtyři BGD.

Pro tuto změnu musíme nejprve nadefinovat jednotlivé časové intervaly. Použijeme následující sled kroků:

1. Spustíme transakci ST04
2. Z menu Operation Mode vybereme položku Timetable
3. Zvolíme Normal operation, dále tlačítko Change
4. Označíme začátek časového intervalu (6:00)

5. Označení potvrdíme přes volby Operation mode > Select interval
6. Označíme konec časového intervalu (22:00)
7. Označení potvrdíme přes volby Operation mode > Select interval
8. Zmáčkne tlačítko Assign pro přiřazení módu
9. Vybereme mód day
10. Opakujeme body 4 až 9 za použití intervalu 22:00 – 6:00 a volby módu night
11. Provedené změny uložíme tlačítkem Save (ikona diskety)

Výsledek by měl vypadat takto:

Start/end time	Name of the active operation mode
00.00 - 01.00	night
01.00 - 02.00	night
02.00 - 03.00	night
03.00 - 04.00	night
04.00 - 05.00	night
05.00 - 06.00	night
06.00 - 07.00	day
07.00 - 08.00	day
08.00 - 09.00	day
09.00 - 10.00	day
10.00 - 11.00	day
11.00 - 12.00	day
12.00 - 13.00	day
13.00 - 14.00	day
14.00 - 15.00	day
15.00 - 16.00	day
16.00 - 17.00	day
17.00 - 18.00	day
18.00 - 19.00	day
19.00 - 20.00	day
20.00 - 21.00	day
21.00 - 22.00	day
22.00 - 23.00	night
23.00 - 00.00	night

Obr. 30. RZ04 – tabulka časů

Tím je hotová první část úkolu. Nyní musíme ještě nastavit vlastnosti jednotlivých operačních módů. Přejdeme do základní obrazovky transakce RZ04 a postupujeme podle následujících instrukcí:

1. Zvolíme Instances/operation modes
2. Dále tlačítko Create new instance
3. Vyplníme políčka Host name, Start Profile a Instance profile
4. Vložená data uložíme tlačítkem Save (ikona diskety)
5. Zkontrolujeme, zda byly profily načteny správně.

Pokud jsou profily načteny správně, musí počty work procesů odpovídat aktuálnímu počtu z transakce SM50. V našem případě přehled souhlasí:

CCMS: Maintain Instance Data

Current settings | Maintain details | Check profile

Installation data
Host name: ivml2005
SAP system number: 00

Start profile
Profile name: START_DVEBMGS42_N4SHOST
Display | Change

Instance profile
Profile name: N4S_DVEBMGS42_N4SHOST
Display | Change

Admin. user for start/stop
User name: []

Number of work processes
According to InstProf

Dialog	10
Background	4
Update	1
Update2	1
Enqueue	1
Spool	1
Total	18

Obr. 31. RZ04 – kontrola načtení profilu

A nyní se dostáváme k finální úpravě jednotlivých módů. Předchozí uložení nastavení vyvolalo dialog, ve kterém ho můžeme provést. Vyplníme postupně oba módy a pomocí tlačítek upravíme počty dialogových a BGD work procesů. Následně vše uložíme přes tla-

čítka Save. Pro úplnost ještě uvedme, že navyšování kteréhokoli typu work procesu je vždy na úkor dialogových, které se adekvátně snižují. Celkový počet work procesů zůstane vždy stejný.

instances and their WP distribution									
Server Name	Instance Profile	Dia	BP	BPA	Spo	Upd	Up2	Enq	Sum
	OP Mode								
ivm12005_N4S_00	N4S_DVEBMCS42_N4SHOST								
	day	10	4	-	1	1	1	1	18
	night	6	8	-	1	1	1	1	18

Obr. 32. RZ04 – detaily operačních módů

Přehled (Obr. 32) zobrazuje počty různých work procesů pro jednotlivé módy. Následně provedeme kontrolu, zdali došlo k přepnutí dialogových work procesů na BGD (označené v přehledu módů jako BP z anglického Batch Process). Počkáme do doby aktivace nočního módu, tj. 22:00 a zkontrolujeme počet work procesů přes transakci SM50 (Obr. 33). Též překontrolujeme, zdali se úlohy již nezpožďují.

Process Overview

No.	Type	PID	Status	Runtime	Report	Cl.	User Names
0	DIA	13981	Running		SAPLTHFB	001	SAP*
1	DIA	13982	Waiting				
2	DIA	13986	Waiting				
3	DIA	13987	Waiting				
4	DIA	13988	Waiting				
5	DIA	13989	Waiting				
6	UPD	13990	Waiting				
7	ENQ	14000	Waiting				
8	BGD	14001	Waiting				
9	BGD	14002	Waiting				
10	BGD	14003	Waiting				
11	BGD	14004	Waiting				
12	BGD	14017	Waiting				
13	BGD	14019	Waiting				
14	BGD	14020	Waiting				
15	BGD	14021	Waiting				
16	SPO	14022	Waiting				
17	UP2	14033	Waiting				

Obr. 33. SM50 – přehled work procesů

Job	Status	Duration	Delay (sec.)
<input type="checkbox"/> CCMS_COLL_DB:RSDB_PAR	Finished	1	0
<input type="checkbox"/> CCMS_COLL_DB:RSDB_TDB	Finished	11	0
<input type="checkbox"/> CCMS_COLL_DB:RSDB_TDB	Finished	6	0
<input type="checkbox"/> DBA:BACKUP_DEV_____@234507/5000	Finished	15	0
<input type="checkbox"/> DBA:REORGCK_ALL_____@214232/4000	Finished	1.166	0
<input type="checkbox"/> RSPARAGENER8	Finished	2.088	0
<input type="checkbox"/> SAP_CCMS_MONI_BATCH_DP	Finished	1	0
<input type="checkbox"/> SAP_CCMS_MONI_BATCH_STARTUP_DP	Finished	31	0
<input type="checkbox"/> SAP_COLLECTOR_FOR_JOBSTATISTIC	Finished	3	0
<input type="checkbox"/> SAP_COLLECTOR_FOR_NONE_R3_STAT	Finished	1	0
<input type="checkbox"/> SAP_COLLECTOR_FOR_PERFMONITOR	Finished	6	0
<input type="checkbox"/> SAP_REORG_BATCHINPUT	Finished	1	0
<input type="checkbox"/> SAP_REORG_JOBS	Finished	0	0
<input type="checkbox"/> SDB_MONITORING_DATA_AUTO_ACTIONS	Finished	2	0

Obr. 34. SM37 – zpoždění úloh

Problém je vyřešen a úlohy se nezpožďují ani v nočních hodinách (Obr. 34).

ZÁVĚR

Cílem naší práce bylo popsat možnosti zvyšování výkonu SAP instancí typu ABAP. V úvodu jsou vysvětleny některé základní pojmy z oblasti systémů SAP, dále pak jednotlivé typy instancí a v neposlední řadě též work procesy. Následně je diskutována optimalizace výkonu a jsou popsány příslušné analýzy, části paměti, parametry a ostatní související informace. Podstatnou část práce zaujmají popisy možností, jak získat informace pro analýzu.

Jsou též diskutovány výhody a nevýhody databáze SAP HANA. Tato naprosto přelomová databáze představuje budoucnost systémů SAP zejména kvůli svému konceptu, perfektní provázanosti se systémy SAP a hlavně řádově rychlejším poskytnutím dat. Nespornou výhodou je fakt, že SAP HANA je napsána přímo pro systém SAP. To umožňuje efektivnější fungování.

Velký důraz byl kladen na praktickou využitelnost této práce. Informace a principy analýz z teoretické části jsou uvedeny do praxe v praktické části. Názorné ukázky konkrétních analýz jsou velkým přínosem pro administrátory, jejichž úkolem je provádět optimalizaci výkonu systémů SAP.

Pro autora byla práce též velkým přínosem, neboť bylo nezbytné osvěžit a prohloubit některé znalosti z dané oblasti.

SEZNAM POUŽITÉ LITERATURY

- [1] FÖSE, Frank, Sigrid HAGEMANN a Liane WILL. *SAP NetWeaver AS ABAP system administration*. 4th ed. Boston: Galileo Press, 2012, 747 p. ISBN 15-922-9411-1.
- [2] *Installation guide: SAP Systems Based on the Application Server ABAP of SAP NetWeaver* [online].[cit. 2015-04-01]. Dostupné z: <http://service.sap.com/instguides>
- [3] *Cluster Technology*. [online].[cit. 2015-03-20]. Dostupné z: http://help.sap.com/saphelp_nw70/helpdata/en/08/5743714ae611d1894f0000e829fbbd/content.htm
- [4] SCHNEIDER, Thomas. *SAP R/3 performance optimization: the official SAP guide*. San Francisco: Sybex, 1999. xxix, 545 s. ISBN 07-821-2563-8.
- [5] *Profile Parameters of Memory Management* [online].[cit. 2015-04-11]. Dostupné z: http://help.sap.com/saphelp_nw73ehp1/helpdata/en/49/3431b15cce5717e10000000a42189b/content.htm
- [6] BREMER, Richard a Lars BREDDemann. *SAP HANA administration*. 1st edition. 722 pages. ISBN 978-159-2299-546

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ABAP	Advanced Business Application Programming
ASCS	ABAP SAP Central Services
ERP	Enterprise Resource Planning
I/O	Input/Output
OS	Operační systém
SAN	Storage Area Network
SQL	Structured query language
TSM	Tivoli Storage Manager

SEZNAM OBRÁZKŮ

Obr. 1. Přehled instancí.....	12
Obr. 2. Přehled work procesů.....	15
Obr. 3. RZ11 – parametr abap/heap_area_total.....	22
Obr. 4. Transakce ST03.....	27
Obr. 5. ST03 – Workload Overview.....	29
Obr. 6. ST03 – Transakční profil.....	32
Obr. 7. ST03 – časový profil.....	33
Obr. 8. Monitor operačního systému.....	36
Obr. 9. Kolektor v ST06.....	37
Obr. 10. Proces kolektoru na OS.....	37
Obr. 11. Transakce SM37.....	39
Obr. 12. Možnosti OS monitoru.....	41
Obr. 13. ST02 – detail bufferu.....	47
Obr. 14. ST03 – Workload overview.....	53
Obr. 15. ST02 – časový profil.....	54
Obr. 16. DB02 – čas zálohy databáze.....	55
Obr. 17. SM37 – přehled úloh.....	55
Obr. 18. ST02 – přehled bufferů.....	56
Obr. 19. Detail program bufferu.....	57
Obr. 20. Volba pro zobrazení parametrů.....	58
Obr. 21. Příslušné parametry.....	58
Obr. 22. RZ11 – parametr abap/buffersize.....	59
Obr. 23. Změna profilu.....	59
Obr. 24. Změna parametru.....	60
Obr. 25. Vyhodnocení změny.....	60
Obr. 26. SM37 – přehled úloh.....	61
Obr. 27. SM50 – přehled work procesů.....	61
Obr. 28. SM37 – čas spuštění.....	62
Obr. 29. SM50 – navýšený počet work procesů.....	63
Obr. 30. RZ04 – tabulka časů.....	64
Obr. 31. RZ04 – kontrola načtení profilu.....	65
Obr. 32. RZ04 – detaily operačních módů.....	66

Obr. 33. SM50 – přehled work procesů.....	67
Obr. 34. SM37 – zpoždění úloh.....	67

SEZNAM TABULEK

Tab. 1. Procesy a jejich účel.....	38
Tab. 2. Význam popisků.....	40
Tab. 3. Sloupce přehledu bufferů.....	45
Tab. 4. Příslušné parametry.....	58