

# **Automatické dohledování a správa systémů s využitím instrumentačních technik**

Automatic Tracing and Management Systems using Instrumental  
Techniques

Bc. Ondřej Hanzal

---

Diplomová práce  
2012



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2011/2012

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Ondřej HANZAL**

Osobní číslo: **A09704**

Studijní program: **N 3902 Inženýrská informatika**

Studijní obor: **Informační technologie**

Téma práce: **Automatické dohledování a správa systémů  
s využitím instrumentačních technik**

Zásady pro vypracování:

1. Zpracujte přehled dostupných nástrojů pro automatické dohledování a správu.
2. Popište vlastnosti, rozdíly, možnosti nástrojů pro automatické dohledování a správu v kontextu rozsáhlých podnikových systémů.
3. Zpracujte přehled technik pro proaktivní predikování chování systému.
4. Vyberte nejvhodnější nástroj pro automatické dohledování a správu a zpracujte jeho nasazení do rozsáhlejšího podnikového systému.
5. Rozšiřte nástroj pro dohledování a správu o správu tiskového řešení a rekonfiguraci systému.
6. Udělejte zhodnocení použitých nástrojů a možnost dalšího rozšíření.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. Nagios Core Version 3.x Documentation [online]. 2010-08-28 [cit. 2012-02-03]. Dostupné z: [http://nagios.sourceforge.net/docs/3\\_0/toc.html](http://nagios.sourceforge.net/docs/3_0/toc.html)
2. Zabbix documentation [online]. 2011-02-16 [cit. 2012-02-03]. Dostupné z: <http://www.zabbix.com/documentation/>
3. Software architecture: 5th European Conference, ECSA 2011, Essen, Germany, September 13-16, 2011. proceedings [online]. 1st ed. New York: Springer, 2011 [cit. 2012-02-03]. ISBN 36-422-3797-5. Dostupné z: [http://books.google.cz/books?id=Z3ngxo-m3xoC&printsec=frontcover&hl=cs&source=gbs\\_ge\\_summary\\_r&cad=0v=onepage&q&f=false](http://books.google.cz/books?id=Z3ngxo-m3xoC&printsec=frontcover&hl=cs&source=gbs_ge_summary_r&cad=0v=onepage&q&f=false)
4. PostgreSQL 8.4 [online]. 2009-07-01 [cit. 2012-02-03]. Dostupné z: <http://www.postgresql.org/docs/8.4/interactive/index.html>
5. PERRY, J. Steven. Java Management Extensions [online]. 1st ed. Cambridge [Mass.]: O'Reilly, c2002, 300 s. [cit. 2012-02-03]. ISBN 05-960-0245-9. Dostupné z: [http://books.google.cz/books?id=rLFkIKnCKGYC&printsec=frontcover&hl=cs&source=gbs\\_g](http://books.google.cz/books?id=rLFkIKnCKGYC&printsec=frontcover&hl=cs&source=gbs_g)

Vedoucí diplomové práce:

**Ing. Radek Vala**

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

**24. února 2012**

Termín odevzdání diplomové práce:

**21. května 2012**

Ve Zlíně dne 24. února 2012



prof. Ing. Vladimír Vašek, CSc.  
*děkan*



doc. Mgr. Roman Jašek, Ph.D.  
*ředitel ústavu*

## **ABSTRAKT**

Práce srovnává komplexní systémy pro dohledování a správu systémů s rozšířením o tisková řešení. Pro každý systém je provedena analýza využitelnosti v rozsáhlých distribuovaných systémech. Je zde definováno obecné tiskové řešení. Praktická část pojednává o návrhu implementace dohledového systému s vybraným tiskovým řešením.

Klíčová slova: ZABBIX, NAGIOS, JAVA, JMX, centrální server, distribuovaný server, SNMP, správa databáze

## **ABSTRACT**

The thesis compares complex monitoring and management system solutions with enhanced printing solutions. It includes analysis of usability in large distributed systems for each system. The author defined a General Printing Solution. The practical part deals with the design implementation of the monitoring system with the selected printing solutions.

Keywords: ZABBIX, NAGIOS, JAVA, JMX, Central Server, Distributed Server, SNMP, Database management

Na tomto místě bych rád poděkoval Ing. Radkovi Valovi za jeho odborné vedení diplomové práce.

**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 s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování 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 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

.....  
podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>11</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>12</b>
<b>1 POJMY Z OBLASTI DOHLEDOVÁNÍ</b> .....	<b>13</b>
1.1 ARCHITEKTURA KLIENT-SERVER .....	13
1.1.1 Server .....	13
1.1.2 Klient.....	13
1.2 CENTRALIZACE A DISTRIBUOVÁNÍ.....	14
1.2.1 Agent .....	14
1.3 SBĚR DAT .....	14
1.3.1 Pooling .....	14
1.3.2 Trapping .....	14
1.4 SNMP.....	15
1.4.1 Protocol Data Units (PDU) .....	15
1.4.2 Object Identifier (OID).....	15
1.4.3 Management Information Base (MIB) .....	16
1.4.4 SNMPv1 .....	17
1.4.5 SNMPv2.....	17
1.4.6 SNMPv3.....	17
1.4.7 Komunita.....	18
1.5 SERVICE LEVEL AGREEMENT (SLA) .....	18
<b>2 NÁSTROJE PRO AUTOMATICKÉ DOHLEDOVÁNÍ A SPRÁVU</b> .....	<b>19</b>
2.1 PŘEHLED .....	19
<b>3 TECHNOLOGIE JMX</b> .....	<b>20</b>
3.1 ARCHITEKTURA.....	21
3.1.1 Distribuční vrstva .....	21
3.1.2 Agentní vrstva .....	22
3.1.3 Instrumentační vrstva .....	23
3.1.4 MBeans.....	23
3.2 MONITOROVÁNÍ A SPRÁVA .....	24
3.2.1 JAVA JConsole.....	24
3.2.2 JAVA VisualVM.....	25
<b>4 NAGIOS</b> .....	<b>26</b>
4.1 VLASTNOSTI.....	26
4.2 PREREKVIZITY .....	26
4.3 KONFIGURACE.....	27
4.3.1 Server .....	27
4.3.2 Klient.....	28

4.4	OBJEKTY .....	30
4.5	SYSTÉM ZÁSUVNÝCH MODULŮ .....	31
4.6	MAKRA .....	33
4.6.1	Uživatelská makra .....	33
4.6.2	Standardní makra .....	34
4.7	SYSTÉM KONTROL .....	35
4.7.1	Aktivní kontroly .....	35
4.7.2	Pasivní kontroly .....	35
4.7.3	Kontrola stanic .....	36
4.7.4	Kontrola služeb .....	37
4.7.5	Typy chybových stavů .....	38
4.8	OBSLUHA UDÁLOSTÍ .....	39
4.8.1	Typy obsluhy událostí: .....	39
4.8.2	Princip obsluhy událostí .....	39
4.9	PŘIDANÉ HODNOTY SYSTÉMU NAGIOS – FLAPPING, BLACKOUT .....	40
4.9.1	Flapping .....	40
4.9.2	Blackout .....	41
4.10	CENTRALIZACE A DISTRIBUOVANÉ MONITOROVÁNÍ APLIKACÍ .....	42
4.10.1	Centrální server .....	42
4.10.2	Distribuovaný server .....	42
4.10.3	Komunikace .....	43
<b>5</b>	<b>ZABBIX .....</b>	<b>44</b>
5.1	VLASTNOSTI .....	44
5.2	PREREKVIZITY .....	45
5.3	KONFIGURACE .....	45
5.3.1	Server .....	45
5.3.2	Proxy .....	46
5.3.3	Klient .....	46
5.4	KOMPONENTY .....	47
5.4.1	ZABBIX Server .....	47
5.4.2	ZABBIX Proxy .....	47
5.4.3	ZABBIX Agent serveru .....	47
5.4.4	ZABBIX Agent .....	48
5.4.5	Utilita pro Unix – ZABBIX Get .....	48
5.4.6	Utilita pro Unix – ZABBIX Sender .....	48
5.5	WEBOVÝ SERVER, ADMINISTRACE A SPRÁVA .....	49
5.5.1	Webový server .....	49
5.5.2	Administrace .....	49
5.5.3	Konfigurace .....	51
5.6	MONITOROVÁNÍ PROVOZU .....	53
5.6.1	Pasivní kontroly .....	53
5.6.2	Aktivní kontroly .....	53

5.7	PŘÍDAVNÉ HODNOTY SYSTÉMU ZABBIX .....	54
5.7.1	Automatické zakládání stanic a služeb.....	54
5.7.2	Opakované oznamování a eskalace.....	55
5.8	CENTRALIZACE A DISTRIBUOVANÉ MONITOROVÁNÍ APLIKACÍ .....	56
5.8.1	Hlavní server (Master node).....	56
5.8.2	Podřízené uzly (ZABBIX servery) .....	56
5.8.3	Podřízené uzly (ZABBIX proxy servery).....	56
5.8.4	Agenti .....	56
<b>6</b>	<b>PROAKTIVNÍ PREDIKOVÁNÍ .....</b>	<b>57</b>
6.1	CO JE PROAKTIVNÍ PREDIKOVÁNÍ.....	57
6.2	JAK FUNGUJE PROAKTIVNÍ PREDIKOVÁNÍ.....	57
6.3	NÁSTROJE PROAKTIVNÍHO PREDIKOVÁNÍ V DOHLEDOVÝCH SYSTÉMECH.....	57
<b>7</b>	<b>TISKOVÁ ŘEŠENÍ.....</b>	<b>58</b>
7.1	ZÁKLADNÍ CHARAKTERISTIKA .....	58
7.2	OBECNÉHO TISKOVÉHO ŘEŠENÍ.....	59
7.2.1	Tisk.....	59
7.2.2	Kopírování.....	62
7.2.3	Skenování .....	64
7.2.4	Faxování .....	64
<b>II</b>	<b>PRAKTICKÁ ČÁST .....</b>	<b>65</b>
<b>8</b>	<b>NÁVRH IMPLEMENTACE PRO TISKOVÉ ŘEŠENÍ YSOFT SAFEQ.....</b>	<b>66</b>
8.1	POŽADAVKY NA MONITOROVACÍ SYSTÉM.....	66
8.2	POROVNÁNÍ VYBRANÝCH MONITOROVACÍCH NÁSTROJŮ .....	68
8.2.1	ZABBIX .....	68
8.2.2	NAGIOS.....	68
8.2.3	JConsole a VisualVM .....	69
8.2.4	Výběr monitorovacího systému.....	69
8.3	NÁVRH IMPLEMENTACE.....	70
8.3.1	Centrální dohledový server .....	70
8.3.2	Podřízené uzly .....	70
8.3.3	Lokální sběrná místa .....	71
8.3.4	Monitorovací agenti .....	71
8.3.5	Webové rozhraní .....	71
8.3.6	Databáze .....	72
8.3.7	Aplikační logika .....	73
8.4	MOŽNOSTI ROZŠÍŘENÍ.....	75
	<b>ZÁVĚR .....</b>	<b>76</b>
	<b>ZÁVĚR V ANGLIČTINĚ.....</b>	<b>77</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>78</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>80</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>81</b>

SEZNAM TABULEK.....	82
SEZNAM PŘÍLOH.....	83

## ÚVOD

S rozvojem počítačových technologií dochází k rozvoji počítačových sítí, podnikových systémů, tiskových řešení. Nové technologie nám přináší změny, které se projevují ve struktuře a správě informačních systémů. S rozrůstajícími se informačními systémy jsou firmy nuceny vynaložit stále více peněz do správy a monitorování těchto systémů. V současnosti je na trhu několik desítek tiskových řešení, které nabízejí různé služby monitorování a správy tiskáren. Navzdory komplexnosti monitorování tiskových zařízení a práce s funkcemi, které tyto systémy nabízejí, pokulhávají v oblasti monitorování sebe sama. Firmy se v současnosti zaměřují na snižování nákladů a v tiskových řešeních vidí cestu, jak zjistit další místa, kde jsou vynakládány zbytečně velké prostředky. Uživatelé mnohdy nemají ponětí, že je v jejich firmě nějaký tiskový systém nasazen a že jejich práce v tomto smyslu je monitorována – například sbírání informací o počtu vytištěných stran za měsíc a názvu dokumentů, které jsou tištěny. S nasazením tiskového řešení do podnikového systému dochází ke zvýšení nároků na údržbu celého systému, jeho správu a monitorování. Instrumentační techniky se zabývají provázaností procesů v systému. Zaměřují se na vyhledání těchto závislostí.

V minulosti bylo monitorování vnímáno především jako podpůrná činnost, která zajišťovala dílčí funkčnosti informačního systému. V současnosti je monitorování úzce spjato s automatizací správy a je vnímáno jako sběr informací o aktuální efektivnosti systému, podklad pro rozšiřování informačního systému, ukazatel plnění stanovených podmínek dostupnosti (SLA) a především zdroj důležitých dat o stavu systému pro efektivní automatickou správu. Nutnost zajištění běhu systému a systém sankcí při částečné či úplné nefunkčnosti tlačí vydavatele softwaru k rozšíření jejich produktu o podporu monitorování a správu. Mnohdy jsou ovšem tyto funkce nedostatečné.

Tato práce se bude v následujících kapitolách zabývat možnostmi monitorování a správou podnikových systémů s tiskovými řešeními. Diskutování obecného tiskového řešení a konkrétního tiskového řešení bude podkladem pro návrh implementace monitorování a správy pro tento druh informačních systémů.

## **I. TEORETICKÁ ČÁST**

## 1 POJMY Z OBLASTI DOHLEDOVÁNÍ

V následující kapitole budou vysvětleny pojmy z oblasti dohledování. Dohledování neboli monitorování, se zabývá sbíráním a vyhodnocováním dat o určitém systému. Systém je v této definici brán jako soubor či soustava navzájem souvisejících elementů, které jsou sdruženy do určitého celku.

### 1.1 Architektura klient-server

„V některých případech, především z důvodu zajištění výkonu systému a jeho bezpečnosti, mohou být programy (komponenty) distribuovány i na více než dva počítače. Tento model se chápe jako víceúrovňová architektura klient-server, kde rozmístěné komponenty serveru kooperují (opět přístupem klient-server) na zajištění zpracování požadavku klienta“[20] Vztah klient-server je založen na komunikaci, kdy klient žádá o určitou službu/informaci server, který mu je poskytuje. Tento vztah má široké uplatnění: např. při přístupu na web, databázi, výběru peněz z bankomatu apod. Zpravidla se jedná o jeden server či skupinu vzájemně spolupracujících serverů a mnoho klientů, kteří se k nim připojují.

#### 1.1.1 Server

- Zpravidla pasivní
- Reaguje na žádosti klientů
- Při přijetí požadavku jej obslouží

#### 1.1.2 Klient

- Zpravidla aktivní
- Posílá žádosti serveru
- Čeká a dostává odpovědi na žádosti
- Komunikuje přímo s koncovým uživatelem

Podle typu závislosti klienta na serveru rozlišujeme tři typy klientů: tlustý, tenký a hybridní. Tlustý klient je připraven na nejhorší, a proto obsahuje „duplicitní“ vlastnosti serveru, tenký klient spoléhá pouze na server a hybridní je kombinace předchozích dvou klientů.

## 1.2 Centralizace a distribuování

Distribuovaný systém je takový systém, který poskytuje uživateli systému dojem, že se jedná o jednotný systém. Ve skutečnosti jde o množinu nezávislých uzlů, které mezi sebou navzájem komunikují pomocí počítačové sítě. Centrální server zpracovává výsledky z monitorování distribuovaných serverů. V nedistribuovaných systémech musí centrální server provádět aktivně monitoring. Distribuované monitorování využívá takzvaných agentů, kteří sbírají data a zasílají je na centrální server, který tyto data vyhodnocuje.

### 1.2.1 Agent

Jedná se o hardware či software, který monitoruje a shromažďuje informace o systému. Tyto informace jsou poskytovány na vyžádání či zasílány automaticky na centrální server ke zpracování.

## 1.3 Sběr dat

### 1.3.1 Pooling

Jedná se o pravidelnou kontrolu systému neboli vzorkování. Klient aktivně sbírá v pravidelných intervalech data, která se vyhodnocují. Používá se např. při kontrole dostupnosti určitého I/O zařízení. Pokud není zařízení připravené, systém čeká. Nevýhodou tohoto sběru dat je „pozdní reakce“ na změnu, která nastala v průběhu vzorkování. Pro rychlejší reakci na změnu je možné zvýšit frekvenci vzorkování. To má ovšem za následek zvýšení zátěže monitorovaného systému.

### 1.3.2 Trapping

Jedná se o využití přerušení, které indikuje změnu stavu hardwaru. Metoda trappingu se využívá například v protokolech SNMP, které zpracovávají asynchronní zprávy. V případě změny je vyslán informační paket tzv. trap. Tento systém obecně není tolik efektivní než pooling.

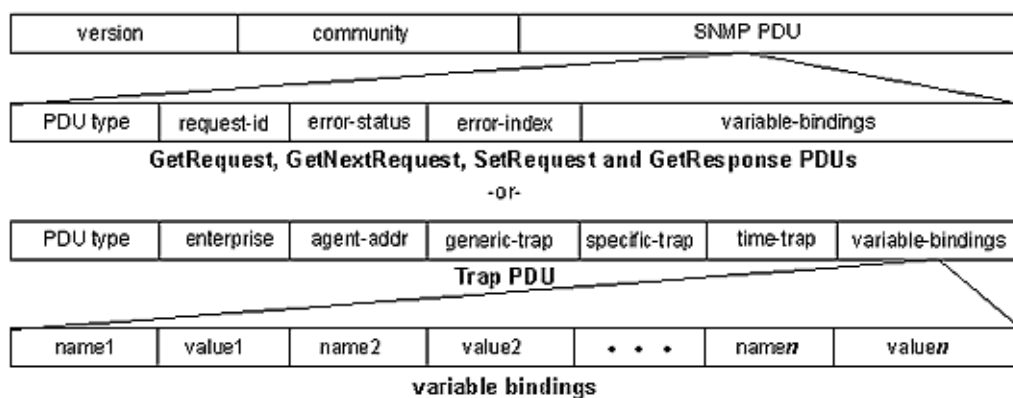
## 1.4 SNMP

Jedná se o jednoduchý, široce rozšířený standardizovaný síťový protokol, který slouží k získávání a nastavování hodnot na zařízení – například síťové prvky, tiskárny, počítačová čidla. Komunikace probíhá mezi stranou manažera (správce) a stranou klienta (agenta). Komunikace může probíhat dvojím způsobem - na jedné straně aktivním dotazováním správce na stranu agenta (již dříve zmíněný pooling) a na druhé straně aktivním zasíláním oznámení o změně (již dříve zmíněný trapping). Pooling může provádět jeden či více správců najednou. Trapping agent může odesílat jednomu či více správcům.

### 1.4.1 Protocol Data Units (PDU)

Jedná se o zprávu, která je odesílána SNMP agentem. Díky jednoduchosti zprávy nedochází k přílišnému zatěžování sítě. Tento typ zpráv je zasílán pouze na vyžádání.

Struktura protokolu je zobrazena na obrázku 1.1.



Obrázek 1.1 SNMP protokol převzato z [3].

### 1.4.2 Object Identifier (OID)

Jedná se o jedinečný identifikátor objektu, která se nachází v informační bázi. OID je tvořeno posloupností čísel oddělených tečkou. Objekty jsou uspořádány do stromové struktury.

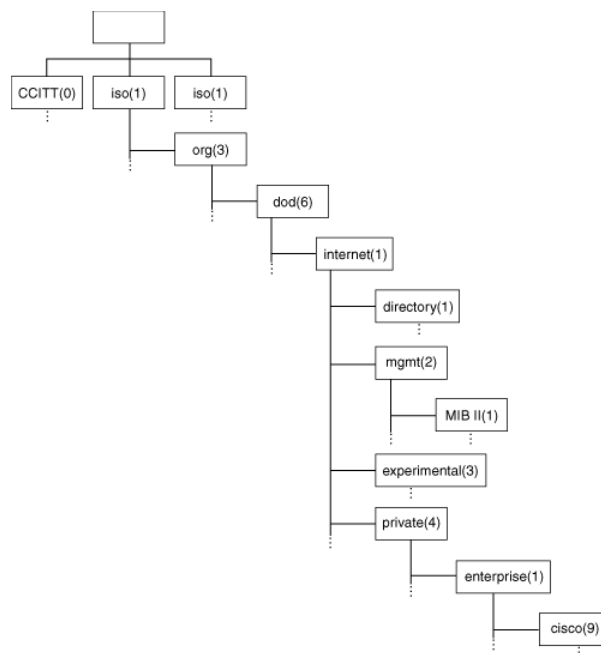
### 1.4.3 Management Information Base (MIB)

MIB stromová forma databáze, která dovoluje jednoznačně identifikovat informace využívané systémem správy. SNMP správce i agent musí znát její strukturu, aby bylo možné informace využít. Báze je objektově orientovaná a její data jsou uloženy jako objekty sdružené do jednotlivých tříd. Stromová struktura báze umožňuje snadnou a přehlednou orientaci.

Každý objekt obsahuje veškeré informace pro popis. Do MIB byly zařazeny pouze nejnütnější objekty. Objekty, které mají určitý aritmetický vztah s MIB objekty, byly vyloučeny, neboť je možné je získat aritmetickými operacemi. Minimalizace MIB zaručuje jednoduchost a nenáročnost agentů, kteří ji využívají. Ukázka stromové struktury MIB je zobrazena na obrázku 1.2. Pro zobrazení struktury MIB můžeme použít například MIB Browser.

Do MIB je možné přidávat objekty pomocí tří mechanismů:

- Pomocí definice nové verze MIB-II
- Pomocí experimentální větve – nestandardní objekty
- Pomocí podstromu soukromé větve – vlastní objekty



Obrázek 1.2 Základní struktura MIB, převzato z [8].

#### 1.4.4 SNMPv1

Jedná se o základní protokol SNMP. Jeho problémem je slabé zabezpečení, které spočívá v ukládání hesel pomocí „community string“. Hesla jsou přenášena nezabezpečeně. Tento protokol podporuje základní operace:

- *GetRequest* – správce vyžaduje informace od agenta
- *GetNextRequest* – správce vyžaduje další položku v MIB stromu
- *GetResponse* – agent posílá odpověď na *GetRequest* správci
- *SetRequest* – správce sděluje agentovi, aby nastavil určitou hodnotu
- *Trap* – agent posílá správci, pokud nastala nějaká předem daní událost

#### 1.4.5 SNMPv2

Tento protokol je založen na předchozím protokolu, který dále rozšiřuje. Nevýhodou je přetrvávající využívání „community stringu“ k ukládání hesel a vzájemná nekompatibilita s předchozím protokolem, která je způsobena změnou formátu zpráv a operací. Protokol obsahuje nové operace:

- *Inform* – správce komunikuje s dalším správcem

V tomto protokolu se je již implementována práce s blokem dat, kterou zprostředkuje typ:

- *GetBulk* – správce požaduje získání větší části MIB stromu najednou

#### 1.4.6 SNMPv3

Nejnovější verze protokol SNMP. Nemění operace předchozího protokolu, ale odstraňuje jeho největší nedostatek – nízké zabezpečení. Tento protokol je uživatelsky orientovaný. Tento protokol nabízí tyto služby:

- *Authentication(autentizace)* – správce je nucen použít přihlašovací jméno a heslo, které je přenášeno v hash formě – použito MD5 a SHA. Předchozí protokoly přenášely tyto údaje v textové podobě
- *Privacy(soukromí)* – zprávy mezi správcem a agentem jsou posílány šifrovaně. Můžeme si vybrat ze symetrických blokových šifer – AES a DES.
- *Access control(řízení přístupu)* – lze omezit přístup správce

### 1.4.7 Komunita

Jedná se o jednoznačný identifikační údaj pro agenta nebo skupinu agentů. SNMP zprávy musí obsahovat název komunity. Základní používané názvy jsou:

- Public – komunita pouze pro čtení
- Private – komunita pro čtení i zápis

Tyto definice lze měnit, ale není to doporučováno, neboť výrobci softwaru obvykle předpokládají defaultní nastavení těchto komunit. SNMP umožňuje definování vlastních komunit, které vedou k zvýšení bezpečnosti a detailnějšímu rozdělení jednotlivých práv agentů a skupin agentů. Komunikace jednotlivých agentů a stanic je povolena pouze v případě, že používají stejnou komunitu. Komunitu bylo nutné používat hlavně ve starších protokolech SNMP. Protokol SNMPv3 umožňuje identifikaci klienta a skupiny klientů pomocí uživatele a hesla. Není tedy explicitně nutné definovat komunity.

## 1.5 Service Level Agreement (SLA)

Jedná se o dohodu o úrovni poskytování služeb. Smlouva se obvykle uzavírá mezi poskytovatelem a zákazníkem (uživatelé). Smyslem této smlouvy je definovat požadovanou úroveň poskytovaných služeb. Součástí smlouvy by měla být jasně stanovaná měření:

- Kvalitativní metriky
- Servisní metriky
- Procesní metriky

Základním cílem tohoto dokumentu je preventivně předcházet tomu, aby nedošlo k chybám při poskytování služeb v důsledku rozdílných očekávání na straně poskytovatele a zákazníka. Preventivní mechanismy jsou klíčové pro zajištění očekávané úrovně služeb a vyhnutí se penále, které vzniká v důsledku jejich neplnění. Tyto penále zpravidla plně nepokrývají veškeré finanční škody, které zhoršená kvalita poskytování služeb způsobí. Definicí kvality poskytování služeb se předchází zbytečným nákladům ze strany poskytovatele i zákazníka a jasně určuje metody jejího měření.

## 2 NÁSTROJE PRO AUTOMATICKÉ DOHLEDOVÁNÍ A SPRÁVU

V současné době existují desítky nástrojů pro monitorování a správu systémů.

### 2.1 Přehled

Nejznámější nástroje pro správu a monitorování můžeme rozdělit na:

1. komplexní
  - ZABBIX
  - NAGIOS
  - IBM Tivoli Network Manager
2. se zaměřením na databáze
  - Microsoft Management Studio
  - Oracle Enterprise Manager
3. se zaměřením na programovací jazyk aplikace, například JAVA (JMX)
  - JConsole
  - VisualVM

V následujících kapitolách se budeme věnovat blíže nástrojům pro monitorování a správu aplikací napsaných v programovacím jazyce JAVA[6] (vybrané tiskové řešení YSoft SafeQ je postavené na programovacím jazyce JAVA) a nástrojům ZABBIX a NAGIOS z pohledu komplexních řešení (ostatní komplexní monitorovací systémy byly zavrženy na základě licenčních ujednání či absence klíčových funkcí pro monitorování rozsáhlých podnikových systému obsahujících tiskový systém).

### 3 TECHNOLOGIE JMX

JAVA jako jeden z mladších, rychle rozvíjejících se programovacích jazyků postupem času potřebovala prostředky pro správu a monitorování aplikací v ní napsaných. Tímto tématem se začala zabývat skupina, která dohlíží na vývoj jazyka a připravuje nové podněty k novým specifikacím – JAVA Community Process (JSP)[5]. Na základě žádostí jednotlivých členů komunity – JAVA Specification Request (JSR)[5], vznikla specifikace JAVA Management eXtensions (JMX)[5]. Přehled hlavních požadavků je zobrazen v tabulce 3.1. JMX bylo zpočátku distribuováno separátně od distribuce samotné Javy. Od verze Javy J2SE5.0 je již JMX standardní plnohodnotnou součástí. Aplikační rozhraní JMX je uloženo v knihovně *JAVAx.management*. Technologie JMX definuje návrhové vzory, aplikační rozhraní, služby pro monitorování a management na platformě JAVA. Už od prvotních požadavků byla JMX navržena tak, aby nově vzniklé JMX systémy mohly být plně integrovány do stávajících systémů. JMX představuje jednoduchou a efektivní cestu, jak rozšířit stávající kód aplikace o nové prvky a tím zpřístupnit její správu a monitorování.

Příklady využití JMX: Webové servery – Apache Tomcat, aplikační servery – JBoss, robustní monitorovací aplikace – ZABBIX, NAGIOS, Hyperic.

JSR	Obsah
JSR000003	Základní specifikace JMX
JSR000077	Specifikace správy J2EE
JSR000151	J2EE 1.4
JSR000160	Vzdálený přístup k JMX
JSR000174	Specifikace monitorování a správy JVM
JSR000176	J2SE 1.5
JSR000255	JMX verze 2.0 a dodatečná funkcionalita
JSR000262	Přístup k managementu pomocí webových služeb

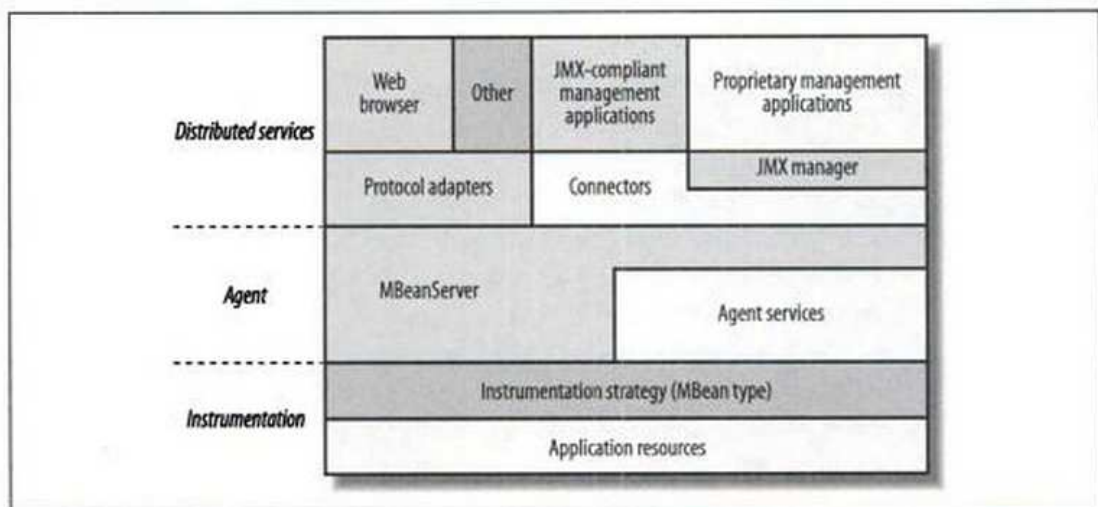
Tabulka 3.1 Tabulka žádostí JSR pro JMX.

### 3.1 Architektura

Architektura JMX se skládá z tří vrstev:

- Distribuční
- Agentní
- Instrumentační

Přehled komponent jednotlivých vrstev je zobrazen na obrázku 3.1.



Obrázek 3.1 Architektura JMX, převzato z [5].

#### 3.1.1 Distribuční vrstva

Distribuční vrstva slouží k poskytování přístupu řídicí aplikaci k JMX Agentovi. Existují dva druhy přístupů: konektory a adaptéry

##### Adaptéry

Poskytují přístup k objektům MBeans prostřednictvím určitých protokolů. Zvolenými protokoly mohou být například HTTP či SNMP protokoly.

##### Konektory

Poskytují přístup k API rozhraní agenta. Používá se například JAVA RMI.

### 3.1.2 Agentní vrstva

Tato vrstva poskytuje aplikacím pro správu a monitorování přístup ke spravovaným a monitorovaným zdrojům. Nachází se mezi distribuční a instrumentační vrstvou. Je tvořena jedním či více JMX agentů. Každý agent je tvořen dvěma částmi. Na jedné straně jde o MBean server, sloužící jako registr MBean, a na druhé o služby JMX agenta. JMX Agenti jsou umístěni na stejném virtuálním stroji jako zdroje, které monitorují. Toto umístění jim umožňuje efektivnější práci díky přímému přístupu.

#### MBean server

Slouží jako registr jednotlivých MBean. Celý proces začíná vytvořením instance MBeany a poté její registrace na serveru. Server vyhodnotí, zda tato instance splňuje nutné požadavky pro úspěšnou registraci (jedinečný identifikátor v rámci serveru), a registraci provede. V případě neúspěchu, není MBeana registrována, což znemožňuje její využití.

#### Služby JMX agentů

Služby jsou další součástí JMX agenta. Každá služba je realizována jako samostatná standardní MBeana. Příklady služeb JMX agenta:

- Služby class loaderu (*Dynamic loading services*)  
Služba umožňuje agentovi nahrávat MBeany ze vzdálených umístění.
- Služby časovače (*Timer services*)  
Služba slouží jako plánovač k odesílání upozornění v předem definovaných intervalech. Tato upozornění mohou být spuštěna jednou či opakovaně.
- Služby monitorování (*Monitoring services*)  
Služba slouží k vyhodnocování jednotlivých hodnot monitorovaných atributů MBean. V případě překročení hraniční hodnoty informuje o této skutečnosti jednotlivé posluchače.
- Služby vztahů (*Relation services*)  
Služba, která se stará o uživatelské vtahy mezi MBeans a jejich konzistencí.

### 3.1.3 Instrumentační vrstva

Na této vrstvě probíhá dodatečná instrumentace všech zdrojů dat, určených pro správu a monitorování, aby byly tyto služby umožněny.

### 3.1.4 MBeans

MBeany (*managed beans*)[5] jsou to speciální typy objektů, které umožňují správu a monitorování zdrojů aplikací. Rozhraní MBean poskytuje:

- Název a typ atributu, který lze číst a modifikovat.
- Název a typ operace, která může být volána.
- Typ oznámení, které může být voláno.

#### Typy MBean:

- Standardní – jsou to JAVA objekty, které odpovídají jednotlivým návrhovým vzorům založených na JAVA Bean modelu. Zajišťují přístup k parametrům prostřednictvím metod *get* a *set*.
- Dynamické – objekty jsou definovány až v průběhu běhu aplikace.
- MXBeans – předdefinovaná sada typů MBean. Příklady jsou uvedeny v tabulce 3.2.

Rozhraní	Popis
ClassLoaderMXBean	Rozhraní pro správu načítání tříd pro JVM.
GarbageCollectorMXBean	Rozhraní pro řízení garbage collectoru JVM.
MemoryMXBean	Rozhraní pro správu paměti JVM.
OperationSystemMXBean	Rozhraní pro správu operačního systému, na kterém běží JVM.
RuntimeMXBean	Rozhraní pro správu běhu JVM.
ThreadMXBean	Rozhraní pro správu vláken JVM.

Tabulka 3.2 Příklad rozhraní MXBeans

## 3.2 Monitorování a správa

Pro monitorování a správu aplikací na platformě JAVA a využití API JMX byly do standardních distribucí Javy přidány aplikace, které tuto funkcionalitu obsahují. Jedná se o aplikace JAVA JConsole a JAVA VisualVM. Druhá ze zmíněných aplikací obsahuje ve své podstatě tu první a další rozšíření na úkor paměťové a procesorové náročnosti na její běh.

### 3.2.1 JAVA JConsole

Je to aplikace s grafickým uživatelským rozhraním, které umožňuje správu a monitorování aplikací na platformě JAVA Standard Edition. Tato aplikace je plně v souladu se specifikací JMX. Jednoduše a efektivně umožňuje monitorování a správu běžících aplikací.

#### Základní komponenty:

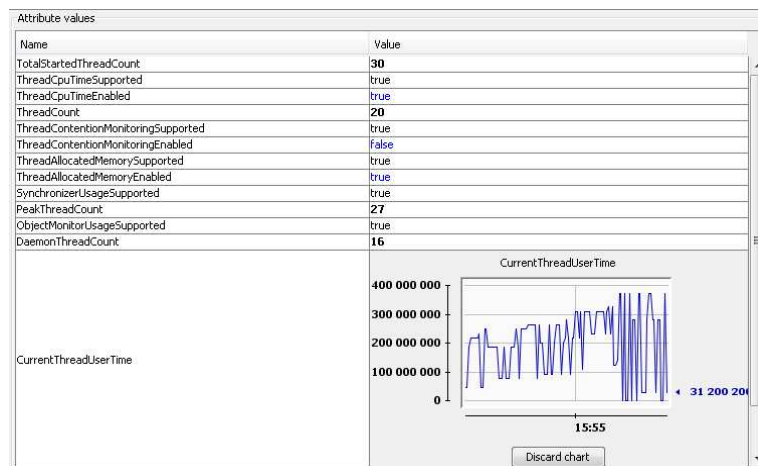
- Přehled – zobrazuje informace o sledovaných hodnotách JVM
- Využití paměti – zobrazuje informace o využití jednotlivých pamětí a práci Garbage collectoru. Umožňuje vynucené zavolání Garbage Collectoru.
- Vlákna – zobrazuje informace o počtu běžících vláken. Umožňuje detekování deadlocků a používání filteru pro výběr určitého typu vlákna. Jednotlivá vlákna obsahují informace o názvu vlákna, délce využívání CPU, chybový výpis.
- Třídy – zobrazuje počet načtených tříd.
- Souhrnné informace o virtuálním stroji
- Strom MBean – zobrazuje veškeré informace o registrovaných MBeans. Umožňuje spravovat jednotlivé MBeans, nastavovat hodnoty parametrů, volat zpřístupněné funkce.
- Další doplňky – JConsole podporuje při startu načtení dalších zásuvných modulů, které slouží k rozšíření správy a monitorování. Většinou se jedná o další záložky, které jsou pomocí zásuvného modulu spravovány. Nejpoužívanějším zásuvným modulem je například JTop, který umožňuje přehledně zobrazit seznam vláken podle přiděleného procesorového času.

### 3.2.2 JAVA VisualVM

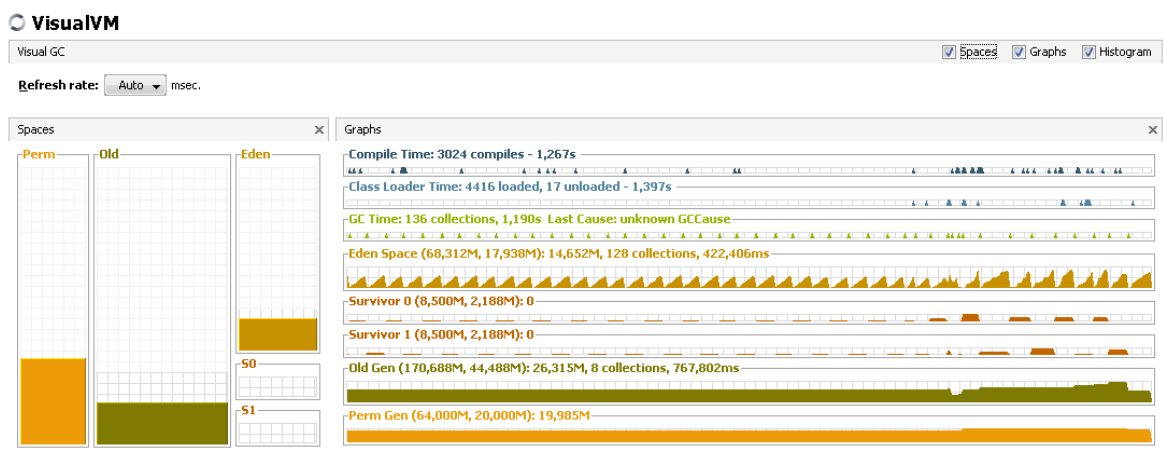
Aplikace pro správu a monitorování aplikací na platformě JAVA Standard Edition.

Poskytuje další funkce oproti JConsoley:

- Snapshoty aplikací, vláken, JAVA heap.
- Podpora zásuvných modulů – k základnímu programu je možné jednoduše doinstalovat: JAVA ME Profiler Snapshot Viewer, VisualVM-Extensions, VisualVM-JConsole, VisualVM-MBeans, Visual GC, Threads Inspector a další.
- Jde o graficky uživatelsky příjemnější prostředí, které zobrazuje detailní pohledy a provázanosti jednotlivých prvků – jako příklad použit Visual GC zásuvný modul na obrázku 3.3 a správa MBean vláken na obrázku 3.2. shodné pro obě aplikace.



Obrázek 3.2 VisualVM - nastavení MBean vláken.



Obrázek 3.3 VisualVM - zásuvný modul Visual GC.

## 4 NAGIOS

NAGIOS[1] je systém pro monitoring systémů a sítí.

### 4.1 Vlastnosti

- Webové rozhraní umožňující zobrazení aktuálního stavu monitorovaných objektů, historii reportovaných hlášení, souborů s logovacími informacemi.
- Monitoring přidělených zdrojů.
- Monitoring síťových služeb.
- Síťová hierarchizace strojů.
- Reportování nedostupnosti, chybovém stavu stroje, výskytu problému i jeho vyřešení.
- Možnost proaktivní reakce na výskyt události na základě handleru událostí.
- Automatická rotace souborů s logovacími informacemi.

### 4.2 Prerekvizity

- Operační systém Linux\Unix.
- Nagios-plugins

Jedná se o skupinu skriptů (může se jednat i o binárních soubory), které samostatně zprostředkovávají vyhodnocení dostupnost určitých služeb a tyto informace dále předat řídicí aplikaci.

- Webový server

Doporučen je webový server Apache s podporou gd knihovny. Tato knihovna je požadována pro zobrazení stavových map a trendů CGI.

- NRPE, NSClient++

NAGIOS AGENT pro monitorování vzdálených stanic s Operačním Systémem Linux\Unix, Windows.

## 4.3 Konfigurace

Konfigurace je uložena v textových souborech, které jsou využívány při startu aplikace. Pro aplikování změny konfigurace je nutné restartovat aplikaci. Před spuštěním po rekonfiguraci je doporučeno provést validaci konfiguračních souborů. V případě výskytu problému je chyba odhalena tímto testem, který zároveň zobrazí pozici výskytu této chyby. Pokud by došlo ke spuštění aplikace před provedením testu s výskytem chyby, je aplikace po jejím nalezení ukončena.

### 4.3.1 Server

Konfigurační soubory serveru se dělí na:

- Hlavní soubor

Soubor je předáván při spuštění aplikace jako parametr. Je využíván NAGIOS serverem i CGI<sup>1</sup>. Obsahuje definice uložení ostatních konfiguračních souborů. Umožňuje povolení či zakázání notifikací, provádění aktivních kontrol, příjmu pasivních kontrol.

- Zdrojové soubory

Jsou používány pro definování maker uživatele. Soubory jsou vhodné pro uložení uživatelských jmen, hesel (je nutné tyto soubory zabezpečit proti zneužití – používá se nastavení práv 600<sup>2</sup> či 660<sup>3</sup>). Slouží k definici databázových připojení.

- Soubory s definicí objektů

Slouží k definování monitorovaných objektů a způsobu jejich monitorování

- CGI konfigurační soubor

Jsou zde uloženy informace o zdrojových souborech webového serveru, které bude rozhraní využívat. Zavádí autentizaci pro přístup na webové rozhraní.

---

<sup>1</sup> Skripty, které generují dynamické webové stránky.

<sup>2</sup> Soubor je čitelný měnitelný pouze jeho vlastníkem

<sup>3</sup> Soubor je čitelný a měnitelný vlastníkem a skupiny uživatelů

### 4.3.2 Klient

Klient systému NAGIOS<sup>4</sup> obsahuje pouze jeden konfigurační soubor. Konfigurační soubor je zcela odlišný od konfiguračních souborů serveru, neboť neobsahuje žádná data.

Hlavními parametry jsou:

- *allowed\_hosts*

Parametr definuje soubor IP adres NAGIOS serverů, od kterých může klient přijímat příchozí spojení. Pokud je parametr nevyplněn, bude NAGIOS klient přijímat tyto spojení bez ověření IP adresy serveru.

- *password*

Definice hesla, které vyžaduje klient po serveru, k vykonání příchozích příkazů. Tento parametr není povinný, ale je doporučený pro zvýšení bezpečnosti.

- *port*

Parametr určuje port NAGIOS klienta. Základní hodnota je 12489[1] pro operační systém Windows a 5666[1] pro operační systém Linux\Unix.

- *modules*

Sekce pro povolení jednotlivých modulů, které bude NAGIOS klient využívat pro monitoring.

- *use\_ssl*

Pokud má server používat šifrované spojení, je nutné tento parametr nastavit na hodnotu 1.

- *NRPE Handlers*

Tyto parametry byly využívány v minulosti, kdy mohlo být na klientu specifikovány vlastní definice kontrol či příkazů, které klient prováděl. V současnosti je tato sekce označena jako „zastaralá“ a neměla by se již používat.

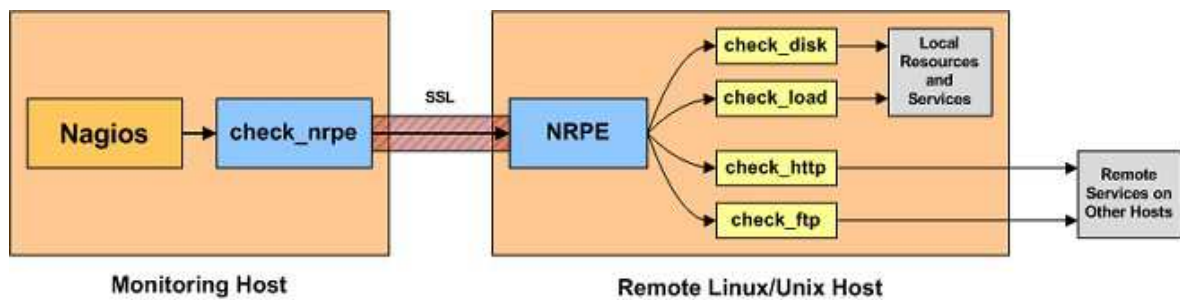
---

<sup>4</sup> NRPE pro operační systémy Linux\Unix, NSClient++ pro operační systémy Windows

## NRPE

Jedná se o NAGIOS klienta, která je instalován na systémy Linux\Unix. Tento klient umožňuje monitorování těchto stanic či stanic, které jsou z nich dostupné.

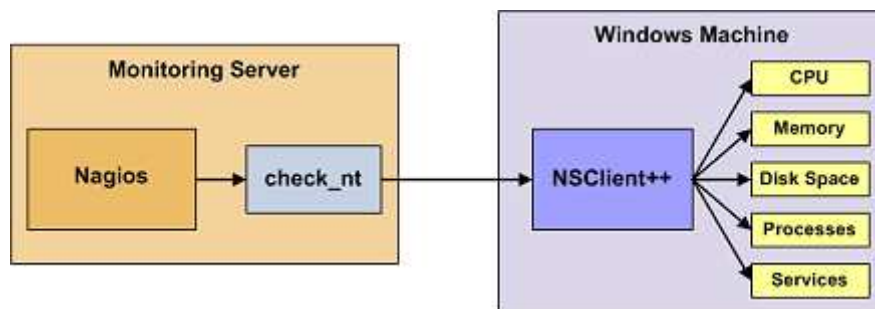
Pro volání příkazů na tomto klientu je použita definice *check\_nrpe*[1]. Příklad volání klienta pro monitorování služeb na klientské stanici pomocí ssl a služeb http a ftp na jiné stanici je znázorněno na obrázku 4.1.



Obrázek 4.1 NAGIOS klient pro operační systém Linux\Unix, převzato z [1].

## NSClient++

Tento klient umožňuje monitorování stanic s operačním systémem Windows nebo stanice, které jsou z nich dostupné. Příkazy pro provedení kontroly pomocí tohoto klienta musí obsahovat definici *check\_nt*[1]. Příklad volání klienta pro monitorování příslušných služeb na klientské stanici je zobrazeno na obrázku 4.2.



Obrázek 4.2 NAGIOS klient pro operační systém Windows, převzato z [1].

## 4.4 Objekty

Objektem se rozumí element, který je využíván aplikační logikou k monitorování a k reakci na určité, předem definované stavy. Dělí se na:

- Služby (*Services*)

Tyto objekty se vztahují k objektu stanice, popisují její vlastnost či službu. Může se jednat například o využití procesoru, volného místa na disku, ale i služby http, ftp.

- Stanice (*Hosts*)

Stroj, na kterém jsou provozovány služby. Jedná se o síťové zařízení (server, tiskárna, atd.). Jsou jednoznačně určeny pomocí IP či MAC adresy.

- Skupiny stanic (*Hosts Groups*)

Díky agregaci strojů do skupin je možné zřehlednit zobrazení jejich stavů na webovém rozhraní a konfiguraci strojů ve skupině.

- Kontakty (*Contacts*)

Adresář uživatelských informací sloužící k doručení oznámení. Jednotlivé kontakty jsou svázány se stroji či službami.

- Skupiny kontaktů (*Contacts Groups*)

Sdružování kontaktů do skupin zjednodušuje definování zasílání oznámení.

- Příkazy (*Commands*)

Definice operací pro volání aplikací využitých pro monitoring či oznámení. Například definice příkazu volání oznámení pomocí emailu: *notify-by-email*.

- Časové periody (*Time Periods*)

Určení časových intervalů pro monitorování a zasílání oznámení kontaktům.

- Oznámení (*Service Escalations, Host Escalations, Hostgroups Escalations*)

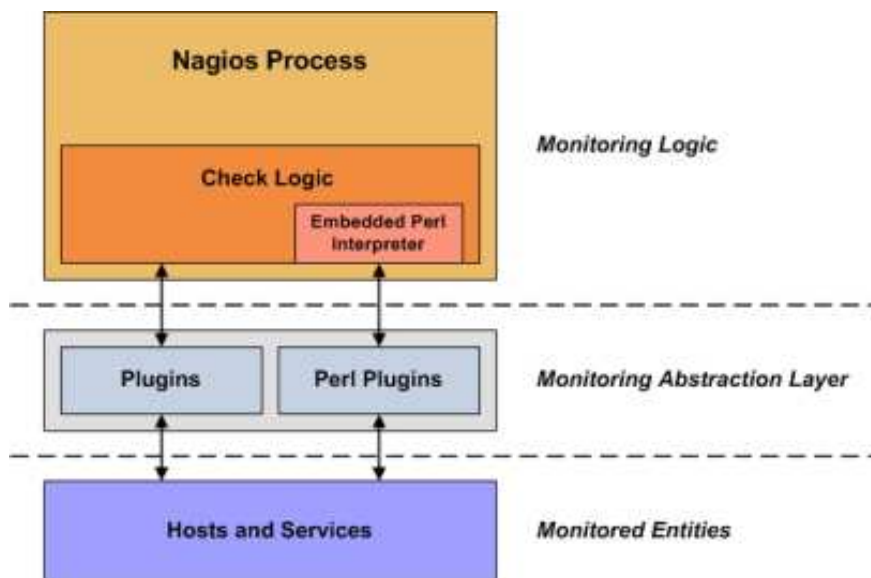
Informování kontaktu o události např. pomocí SMS, emailu, zprávy na pager, aj.

- Závislosti (*Service Dependencies, Host Dependencies*)

Definování jednotlivých vztahů mezi stanicemi, službami (rodič – potomek, aj.).

## 4.5 Systém zásuvných modulů

Zásuvné moduly jsou příkazy nebo skripty, které slouží systému NAGIOS pro provádění kontroly stavů monitorovaných objektů. Systém fungování zásuvných modelů je zobrazen na obrázku 4.3.



Obrázek 4.3 Zásuvné moduly systému NAGIOS, převzato z [1].

NAGIOS neobsahuje žádné interní mechanismy, které by zajišťovaly kontrolu. Monitorovací logika je plně založena na volání zásuvného modulu jako abstraktní monitorovací vrstvy. Systém abstraktní vrstvy odprošťuje samotné jádro NAGIOSu od nutnosti znát vnitřní mechanismy fungování daného objektu a systém provádění jednotlivých dílčích kontrol. Tato logika je obsažena přímo v zásuvném modelu a systému NAGIOS umožňuje volání jednotlivých „funkcí“.

Zásuvné moduly jsou využívány vždy, pokud je nutné zjistit stav daného objektu. Systém NAGIOS po volání zásuvného modulu zpracuje výstupní data, které zásuvný modul po kontrole vrátí vyhodnocení stavu. Pokud jsou na jednotlivé stavy vázány příkazy, jsou tyto příkazy vykonány po vyhodnocení.

### Zásuvné moduly umožňují monitorovat například:

- Sledování síťových služeb (pomocí TCP portů, SMTP, POP3, HTTP, PING, FTP)
- Sledování vytížení CPU, využití HDD, paměti
- Sledování síťových prvků: Router, Switch

System NAGIOS obsahuje vestavěný interpret Perl (ePN)[1], který umožňuje provádět spouštění zásuvných modulů v jazyce Perl. System volání zásuvných modulů je zobrazen na obrázku 4.4.

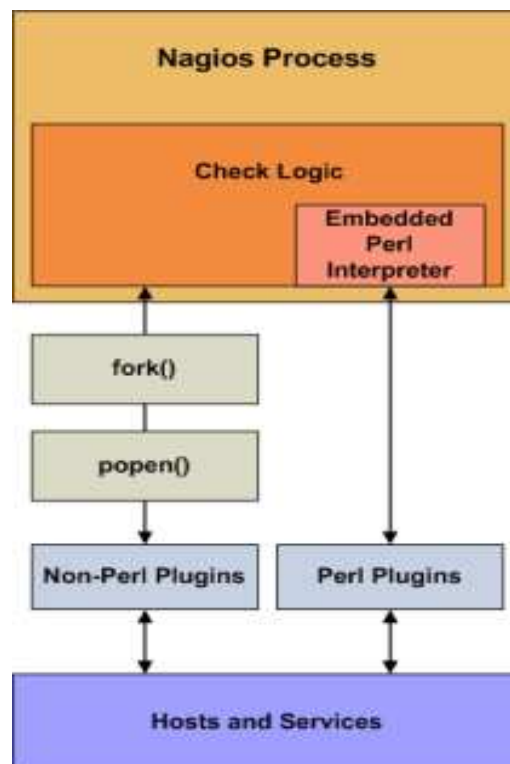
#### Výhody ePN:

- Efektivnější spouštění perl modulů jako funkcí knihovny.
- Zkrácení času spouštění a vykonávání příkazů zásuvného modulu.
- Snížení zátěže systému.

#### Nevýhod ePN:

- ePN je odlišný od standardního jazyka Perl a je nutné použít jinou syntaxi.
- Ladění skriptů interpretovaných ePN je mnohem zdouhavější a náročnější.
- Zvýšení paměťových nároků na běh těchto zásuvných modulů.

Ostatní moduly jsou nazývány jako externí skripty, jak je zobrazeno na obrázku 4.4.



Obrázek 4.4 Volání zásuvných modulů, převzato z [1].

## 4.6 Makra

Makra slouží ke zjednodušení konfigurace systému NAGIOS. Jsou uložena ve zdrojových souborech (kapitola 4.3). Tyto zabezpečené soubory pomáhají skrýt citlivé informace, jako jsou například hesla, výsledky kontrol nebo cesty k jednotlivým souborům. Pokud tyto citlivé údaje nebudeme mít v těchto zabezpečených souborech, zpřístupníme údaje démonu CGI, který je bude moci přečíst a zobrazit. Vytváření maker umožňuje vytvoření proměnných, na které se použijí pro volání jednotlivých příkazů. Před spuštěním konkrétního příkazu bude makro nahrazeno skutečnou hodnotou. Makra mohou být jednoduchá, ale i vnořená (makro obsahuje ve své definici další makro).

Základní dělení maker:

- Uživatelská makra.
- Standardní makra (standardní makra uložená v systému NAGIOS).
- On-Demand makra (Slouží pro předání hodnoty příkazu z jiné stanice, než pro kterou byl tento příkaz spuštěn).

### 4.6.1 Uživatelská makra

Tyto makra si uživatel sám definuje. Jsou uložena v konfiguračním souboru `resource.cfg`. Těchto souborů může být více. Systém NAGIOS umožňuje vytvoření až 256 uživatelských maker (`$USER1$` až `$USER256$`).

Příklad definice uživatelského makra pro definici parametrů zásuvného modulu `check_nt`:

```
$USER1$=/usr/local/nagios/libexec/check_nt (definice desty k zásuvnému modulu)
```

```
$USER2$=54321 (definice portu)
```

```
$USER3$=MojeTajneHeslo (definice hesla)
```

Příklad definice použití předchozího makra:

```
define command{  
    command_name    check_nt  
    command_line    $USER1$ -H 192.168.20.10 -p $USER2$ -s $USER3$  
}
```

#### 4.6.2 Standardní makra

Makra mohou být použita do všech typů příkazů, ovšem makro pro určité typy příkazů nemusí platit (nebude vyhodnoceno).

Rozdělení maker podle toho, pro jaký objekt se používají[1]:

- Makra pro stanici a skupinu stanic.
- Makra pro službu a skupinu služeb.
- Makra pro kontakt a skupinu kontaktů.
- Makra pro sumarizaci.
- Makra pro oznámení.
- Makra pro vyjádření času.
- Makra pro práci se soubory.
- Ostatní makra.

Rozdělení maker podle toho, pro jakou funkci slouží[1]:

- Kontrola služeb a stanic.
- Upozornění služby a stanice.

Obecně se makra pro služby nemohou použít pro kontrolu stanic či upozornění stanice, ale naopak to neplatí. Při kontrole služby můžeme využít makro pro stanici, neboť na stanici služba běží. Makra pro kontakty se využívají pro posílání upozornění a upozornění se nepoužívají pro kontroly.

Příklad použití standardního makra:

```
define command{  
  
    command_name    check_ping  
  
    command_line  /usr/local/nagios/libexec/check_ping -H $HOSTADDRESS$  
  
}
```

## 4.7 Systém kontrol

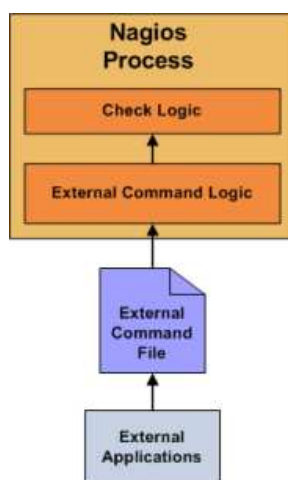
Systém NAGIOS provádí kontroly aktivní a pasivní stanic a služeb.

### 4.7.1 Aktivní kontroly

Tento typ kontrol je nejběžnější metodou pro monitorování stanic a služeb. Aktivní kontroly jsou inicializovány procesem NAGIOSu nebo pomocí pravidelných naplánovaných procesů. Vždy, když je potřeba zjistit stav stroje či služby, dojde ke spuštění zásuvného modulu, který provede kontrolu stavu. Pravidelné kontroly jsou popsány v kapitole 1.3.1.

### 4.7.2 Pasivní kontroly

Tento typ kontrol spoléhá na vyčkávací taktiku. Systém NAGIOS ani jeho agent se nesnaží provádět jakoukoli kontrolu, ale čeká na obdržení dat (výsledků kontrol) externí aplikace, která tyto kontroly provádí za něj. Pasivní kontroly se využívají především pro monitorování služeb, které jsou asynchronní. Není tedy přínosné provádět pravidelné kontroly a zatěžovat tak systém a počítačovou síť. Systém pasivních kontrol je zobrazen na obrázku 4.5. Externí aplikace provede kontrolu a zapíše výsledek této kontroly do souboru, který je pravidelně kontrolován systémem NAGIOS. Při dalším cyklu čtení tohoto souboru dojde k zařazení výsledků těchto kontrol do fronty pro další zpracování. Tato fronta se využívá pro oba typy kontrol – pasivní i aktivní. NAGIOS postupně zpracovává data ve frontě a na jejich základě reaguje.



Obrázek 4.5 Pasivní kontroly, převzato z [1].

### 4.7.3 Kontrola stanic

Stav stanice může nabývat těchto hodnot[1]:

- UP
- DOWN
- UNREACHABLE

Zásuvné moduly, které provádějí kontrolu stanice, pracují s hodnotami: OK, WARNING, UNKNOWN či CRITICAL[1]. NAGIOS vyhodnocuje hodnotu předanou zásuvným modulem a tuto hodnotu převádí na stav stanice, kterou označí jako předběžný. Systém předběžného vyhodnocení je zobrazen v tabulce 4.1. Pro určení konečného stavu stanice se použije další kontrola. Systém vyhodnocení konečného stavu je zobrazen v tabulce 4.2.

Hodnota vrácená zásuvným modulem	Předběžný stav nastavený v NAGIOSu
OK	UP
WARNING <sup>5</sup>	UP nebo DOWN
UNKNOWN	DOWN
CRITICAL	DOWN

Tabulka 4.1 Předběžný stav stanice.

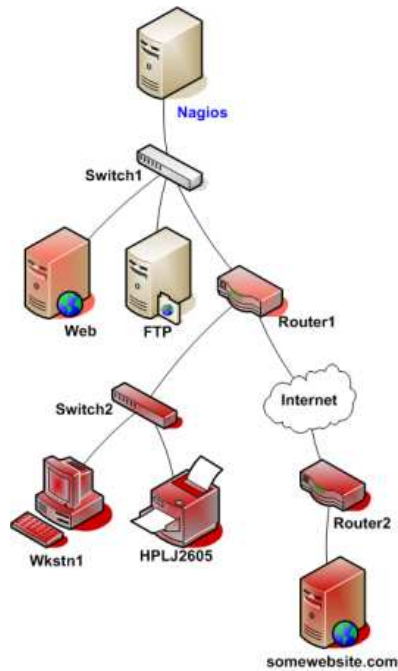
Předběžná hodnota	Stav rodiče	Konečný stav
DOWN	Alespoň jeden je UP	DOWN
DOWN	Všichni jsou DOWN či UNREACHABLE	UNREACHABLE <sup>6</sup>

Tabulka 4.2 Konečný stav stanice

<sup>5</sup> Stav WARNING většinou znamená, že stanice je ve stavu UP. Můžeme ovšem zpřesnit tuto kontrolu parametrem, který nám v konečném vyhodnocení vrátí stav DOWN (například časová dostupnost stanice).

<sup>6</sup> Stav nedostupnosti je zjišťován na základě stavu jeho rodičovských uzlů. Tento stav pomáhá určit rychleji příčinu výpadku stanic.

Praktický příklad vyhodnocení konečného stavu pro síťovou topologii zobrazenou na obrázku 4.6. Web i Router1 jsou vypnuté. Jejich konečný stav bude DOWN, neboť Switch1 je ve stavu UP. Jednotlivé stanice, které se nacházejí v síťové topologii pod stanicí Router1 budou mít v NAGIOSu stav UNREACHABLE, neboť splňují podmínku, že všechny rodičovské uzly jsou v konečném stavu DOWN nebo UNREACHABLE.



Obrázek 4.6 Síťová topologie závislosti stanic a služeb pro vyhodnocení stavu DOWN a UNREACHABLE, převzato z [1].

#### 4.7.4 Kontrola služeb

Stavy služeb vrácené zásuvným modulem a konečné stavy v systému NAGIOS jsou totožné. Jedná se o stavy[1]:

- OK
- WARNING
- UNKNOWN
- CRITICAL

Systém NAGIOS reaguje na tyto hodnoty stavy na základě přiřazení hodnot stavů k typům stavů.

#### 4.7.5 Typy chybových stavů

System NAGIOS rozlišuje dva typy chybových stavů: SOFT a HARD. Typy stavů se používají pro zpracování událostí spojené s doručením upozornění tak, aby nedocházelo k falešným poplachům. System NAGIOS umožňuje specifikovat počet nutných kontrol pro potvrzení změnu typu stavu a rozpoznání skutečného poplachu. K tomuto nastavení se používá konfigurační volba: *max\_check\_attempts*. Výsledky pasivních kontrol mají vždy chybový stav HARD, pokud není povolena konfigurační volba *passive\_host\_checks\_are\_soft*. Přejechod mezi chybovými stavy je zobrazen v tabulce 4.3.

##### Chybový stav SOFT:

- Pokud konečným stavem není OK nebo UP a zároveň nedošlo k vyčerpání všech pokusů pro kontrolu planého poplachu.
- Pokud dojde k zotavení (opětovných stav OK či UP) po předchozím chybovém stavu SOFT.

##### Chybový stav HARD:

- Pokud konečným stavem není OK nebo UP a zároveň nedošlo k vyčerpání všech pokusů pro kontrolu planého poplachu.
- Pokud dojde k zotavení (opětovných stav OK či UP) po předchozím chybovém stavu HARD.
- Pokud dojde ke změně typu stavu (WARNING, CRITICAL, DOWN, UNREACHABLE[1]) po předchozím chybovém stavu HARD.

Čas	Kontroly	Stav	Typ stavu
1	1	OK	HARD
2	1	CRITICAL	SOFT
3	2	WARNING	SOFT
4	3	CRITICAL	HARD
5	1	OK	HARD

Tabulka 4.3 Změny Hard a Soft stavů.

## 4.8 Obsluha událostí

Obsluha události umožňuje Systému NAGIOS proaktivní jednání pro nápravu chybového stavu stanice či služby. Cílem je oprava chybového stavu před zasláním poplašného upozornění. Jedná se o skripty nebo spustitelné soubory, které jsou volány při přechodech stavů SOFT a HARD, nebo při zotavení z chybového stavu.

### 4.8.1 Typy obsluhy událostí:

- Globální pro stanice či služby.
- Globální servisní.
- Specifické pro konkrétní stanici či službu.

### 4.8.2 Princip obsluhy událostí

Systém NAGIOS automaticky definuje, kdy je spuštěna obsluha události. Před spuštěním provádí standardně kontroly podle nastavení o maximálním počtu kontrol před chybovým stavem HARD. Pokud stanice či služba jsou již v chybovém SOFT stavu a proběhl předposlední pokus před stavem HARD, je spuštěn skript či spustitelný soubor definovaný pro obsluhu události. Další spuštění obsluhy události je po prvním přechodu do chybového stavu HARD, pokud stav stanice či služby není OK či UP.

Příklad definice obsluhy události ftp, která provede příkaz pro restartování služby:

```
define service{  
    host_name                localhost  
    service_description      ftp  
    max_check_attempts       5  
    event_handler            restart-ftp  
}  
  
define command{  
    command_name             restart-ftp  
    ...}
```

## 4.9 Přidané hodnoty Systému NAGIOS – flapping, blackout

### 4.9.1 Flapping

Jedná se o detekci velké frekvence změn stavů služby či stanice. Vysoká frekvence těchto změn může být například důsledkem stability síťového připojení či nastavením nízkých prahových hodnot pro kontrolu. Systém NAGIOS umožňuje zapnutí detekce flappingu pomocí konfigurační volby *enable\_flap\_detection*[1]. Systém detekce flappingu spočívá v ukládání výsledků posledních kontrol (21 kontrol) a zaznamenávání změny stavů. Na základě změn je spočítána míra změny stavu (procentuální hodnota změny stavu). Služba či stanice, která má míru změny stavu nulovou, neměnila v posledních 21 kontrolách svůj stav a je tedy absolutně stabilní (služba či stanice může být stabilní i pokud má jiný stav než UP či OK). Algoritmus pro výpočet míry změny stavu pracuje s váhovým systémem hodnot (měnám u mladších hodnot přiděluje vyšší váhu). Výsledná hodnota je podle typu objektu (služba či stanice) porovnána s prahovými hodnotami v konfiguračním souboru (*low\_service\_flap\_threshold*, *high\_service\_flap\_threshold*, *low\_host\_flap\_threshold*, *high\_host\_flap\_threshold*[1]). Nízké prahové hodnoty slouží k vyhodnocení ukončení flappingu a vysoké k jeho startu. Operace při začátku a konci flappingu jsou znázorněny v tabulce 4.4.

Operace	Start flappingu	Stop flappingu
1	Zalogování informace o startu flappingu	Zalogování informace o konci flappingu
2	Přidání komentáře ke službě či stanici o flappingu	Smazání komentáře ze služby či stanice o flappingu
3	Zaslání upozornění o začátku flappingu	Zaslání upozornění o ukončení flappingu
4	Potlačení dalšího zasílání upozornění na stavy pro tuto stanici či službu	Zrušení potlačení zasílání upozornění na stavy pro tuto stanici či službu

Tabulka 4.4 Operace flappingu

### 4.9.2 Blackout

Jde o označení odstávky, kdy chceme provést určitý plánovaný zásah, který ovlivní po určitou dobu funkcionalitu systému<sup>7</sup>. Doba, po kterou je zapnuta odstávka, se nepřipočítává do SLA jako výpadek systému a neposílají se upozornění na stavy služeb a stanic (administrátoři dostanou upozornění na začátku a na konci odstávky).

#### Typy odstávek:

- Flexibilní

Používá se pro definování časové délky odstávky a časového rozmezí, kdy k odstávce dojde. Systém NAGIOS spustí odstávku, jakmile se služba v nastaveném intervalu přepne do chybového stavu.

- Fixní

Používá se pro definování pevného startu a konce odstávky. Start blackoutů začne neohledně na stavu služby či stanice (může být ve stavu OK či UP).

---

<sup>7</sup> Pokud při odstávce zjistíme, že plánovaná doba odstávky nedostačuje pro vykonání zásahu, můžeme vytvořit další odstávku, které se bude překrývat s původní odstávkou.

## 4.10 Centralizace a distribuované monitorování aplikací

Hlavní myšlenkou je rozložení zátěže na ostatní servery a vybudování centrálního serveru pro ukládání všech dat a webového přístupu, jak je znázorněno v příloze P I. Distribuované monitorování není nezbytně nutné pro menší a středně velké podnikové systémy, neboť počet stanic a služeb je relativně malý. U velkých podnikových systémů je již distribuované monitorování nezbytností. Systém NAGIOS dokáže pomocí pasivních kontrol sbírat data od distribuovaných serverů. Pro přenos těchto dat mezi jednotlivými distribuovanými servery směrem k centrálnímu serveru se používá NSCA (Nagios Service Check Adaptor) ve formě klienta a démona.

### 4.10.1 Centrální server

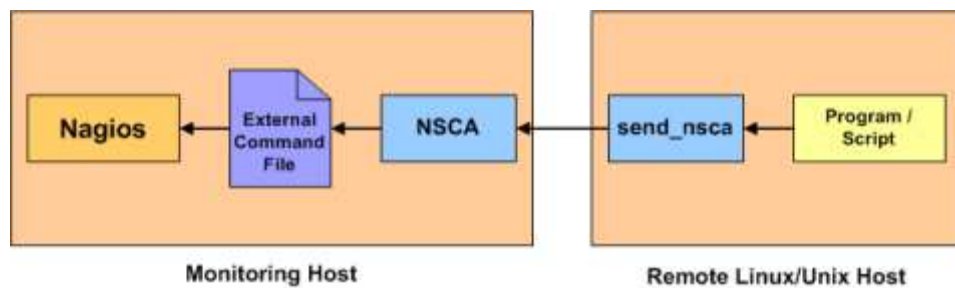
- Obsahuje webové rozhraní.
- Provádí zasílání oznámení.
- Může provádět aktivní či servisní kontroly, ale není to příliš žádoucí.
- Provádí pasivní kontroly.
- Obsahuje všechny definice stanic a služeb, které jsou monitorovány distribuovanými servery.
- Provádí obsluhu událostí.
- Rozhoduje o stavech stanic a služeb.

### 4.10.2 Distribuovaný server

- Instalace na Linux\Unix operační systém.
- Dalo by se říci, že se jedná o kostru NAGIOS serveru.
- Neobsahuje webové rozhraní.
- Neprovádí zasílání oznámení.
- Má zapnuté distribuované monitorování (`obsess_over_services`, `obsess_over_host`).
- Využívá specifických příkazů, které se provádějí po každé servisní kontrole (`ocsp_command` a `ochp_command`).

### 4.10.3 Komunikace

System předávání informací je založen na odesílání dat voláním volání `ocsp` a `ocsh` příkazů, které předávají data NSCA klientů. Tito klienti komunikují s NSCA démonem, která běží na centrálním serveru – zobrazeno v příloze P I. NSCA klient využívá příkazu `send_nsca`[1] pro fyzické předání dat démonovi – zobrazeno na obrázku 4.7. Démon po přijmutí od NSCA klienta provede zápis přijatých dat do externího souboru, který je pravidelně kontrolován Systémem NAGIOS. Data jsou poté zpracovány formou pasivním kontrol popsaných v kapitole 4.7.2.



Obrázek 4.7 Komunikace distribuovaného serveru s NSCA démonem, převzato z [1].

## 5 ZABBIX

System ZABBIX[2] je open-source (licence GPL General Public License version 2) distribuované řešení pro monitorování síťových prvků. Jde o velice oblíbený a rozšířený systém, které je vhodný i pro monitorování rozsáhlých podnikových systémů. Využívá centralizované správy pomocí webového rozhraní a distribuovanou konfiguraci pro získání jednotlivých monitorovacích voleb od agentů směrem k centrálnímu serveru. Tento monitorovací systém se rychle rozvíjí - poslední verze byla vydána v tomto roce (2012).

### 5.1 Vlastnosti

- Webové rozhraní umožňuje zobrazení všech dat jako jsou statistiky, konfigurační parametry, práva uživatelů.
- Pro usnadnění konfigurace využívá systému šablon, na které jsou vázány jednotlivé definice monitorovaných objektů a práce s nimi.
- Využívá relační databáze pro ukládání konfiguračních voleb a výsledků monitorování.
- Monitorování přidělených zdrojů.
- Monitorování síťových služeb.
- Síťová hierarchizace strojů.
- Umožňuje využít agenty pro distribuované monitorování.
- Monitorování SNMP v1, v2, v3.
- Upozornění uživatelů zasílá v podobě emailu, SMS, komunikací serverů (například jabber).
- Obsahuje kontrolu a monitorování SLA.
- Umožňuje zakoupení komerční podpory – nabízí se 6 typů podpory.
- Metodě nejmenších čtverců bude v následující verzi finálně implementována pro predikční systém.

## 5.2 Prerekvizity

- Operační systém Linux\Unix s knihovnamy OpenIPMI, libssh2, fping.
- Relační databáze

Jako defaultní databáze je integrována databáze MySQL. Podporovány jsou dále relační databáze: IBM DB2, PostgreSQL, Oracle, SQLite[2].

- Webový server a PHP

Doporučen webový server Apache od verze 1.3.12. PHP podporováno od verze 5.0 s integrovaným modulem GD 2.0 a novějším. Tento modul se využívá pro zobrazení grafického formátu PNG.

## 5.3 Konfigurace

Hlavní část konfigurace je uložena v relační databázi (bude popsáno v následujících kapitolách). Parametry, které jsou nutné pro start aplikace, jsou uloženy v konfiguračních souborech. Pro aplikování změny konfigurace v konfiguračním souboru je nutné restartování aplikace.

### 5.3.1 Server

Server obsahuje dva konfigurační soubor *ZABBIX\_server.conf* a *ZABBIX.conf.php*.

*ZABBIX\_server.conf*

- Obsahuje připojení na relační databázi (proměnné začínají DB – server, databáze, uživatel, heslo, port, atd.)
- Definuje cesty k logovacím souborům, externím skriptům, dočasným uložištím.
- Specifikuje práci s daty (čas provádění optimalizace, maximální počet uložených hodnot, atd.).
- Vymezuje přidělitelné systémové prostředky pro běh serveru.

*ZABBIX.conf.php*

- Obsahuje připojení na relační databázi a ZABBIX server (je teoreticky možné provozovat server na jiné stanici, než je provozována databáze a webové rozhraní).

### 5.3.2 Proxy

ZABBIX Proxy obsahuje jeden konfigurační soubor *ZABBIX\_proxy.conf*.

Mezi důležité parametry uložené v konfiguračním souboru patří parametry:

- Mód proxy – aktivní či pasivní.
- Připojení na relační databázi, do které jsou ukládány výsledky kontrol.
- Připojení na ZABBIX server.
- Definice času a počtu uložených dat, které čekajících na odeslání ZABBIX serveru.

### 5.3.3 Klient

Klient obsahuje jeden konfigurační soubor v závislosti na typu klienta. Pokud se jedná o klienta, který nativně běží na ZABBIX serveru a monitoruje tento server, jedná se o konfigurační soubor *ZABBIX\_agent.conf*. Dalším případem je standardní agent, který je použit pro monitorování vzdálených stanic a služeb, které na nich běží. Pro tohoto klienta se používá konfigurační soubor *ZABBIX\_agentd.conf*.

*ZABBIX\_agent.conf*

- Obsahuje pouze zlomek nastavení, která jsou v plném klientovi.
- Vymezuje připojení k ZABBIX serveru.
- Obsahuje uživatelské parametry pro monitorování ZABBIX serveru.

Předdefinované jsou parametry pro sledování relační databáze MySQL pro ukládání konfigurací a dat ZABBIX serveru. Tyto parametry jsou označeny jako komentáře, pokud je ZABBIX server nainstalován s podporou jiného typu relační databáze.

*ZABBIX\_agentd.conf*

- Obsahuje připojení na ZABBIX server pro komunikaci.
- Vymezuje velikost zásobníku s daty, která se budou posílat na server.
- Povoluje či zamezuje pasivní a aktivní kontroly či volání vzdálených příkazů.
- Součástí jsou uživatelské parametry pro monitorování ZABBIX serveru

Předdefinované parametry zde chybí.

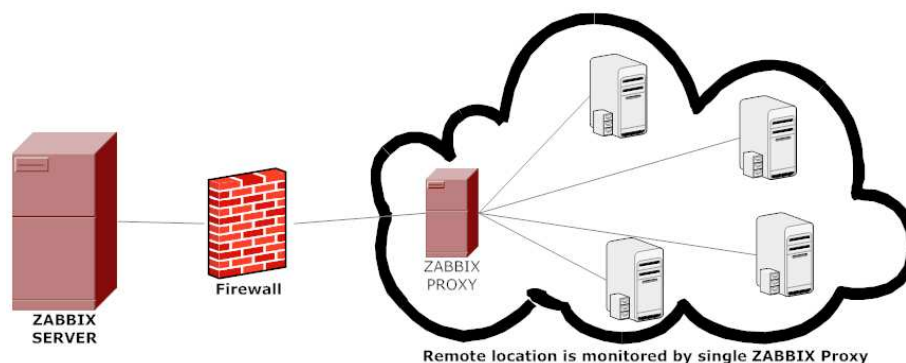
## 5.4 Komponenty

### 5.4.1 ZABBIX Server

ZABBIX server je hlavní částí systému. Server slouží jaké centrální uložení konfiguračních voleb a monitorovaných dat. Může být využit také k monitorování stanic a služeb, ale většinou se k tomuto úkolu využívají jednotliví agenti. Obsahuje webové rozhraní, které umožňuje přehledné zobrazení informací, definování konfiguračních voleb a jednotlivých monitorovaných objektů.

### 5.4.2 ZABBIX Proxy

Jde o volitelnou část systému ZABBIX, která umožňuje zastoupení hlavního serveru ve sběru dat na distribuovaných lokacích, ze kterých by jednotliví klienti mohli mít problém s odesláním dat. Data jsou uložena do lokální relační databáze, ze které jsou poté posílány na hlavní server – zobrazeno na obrázku 5.1. Tato komponenta umožňuje rozložení využití systémových prostředků, neboť odpoštuje hlavní server od části aktivit.



Obrázek 5.1 ZABBIX Proxy, převzato z [2].

### 5.4.3 ZABBIX Agent serveru

Využívá se pro monitorování ZABBIX serveru. Jde o zjednodušeného klienta určeného pouze pro monitorování ZABBIX serveru, na kterém zároveň běží. Vzhledem k odděleným komponentám webového serveru a ZABBIX serveru může administrátor sledovat stav této klíčové komponenty.

#### 5.4.4 ZABBIX Agent

Agent slouží k monitorování lokálně dostupných zdrojů (využití paměti, zaplnění disku, přidělení procesoru) a služeb. Pro sběr dat musí na těchto stanicích běžet. Informace, které získá, předává hlavnímu serveru, který je poté vyhodnotí. Využívá nativních volání pro větší optimalizaci běhu. ZABBIX Agent je podporován pro tyto platformy:

- AIX
- FreeBSD
- HP-UX
- Linux
- Mac OS/X
- NetBSD
- OpenBSD
- SCO Open Server
- Solaris
- Windows

#### 5.4.5 Utilita pro Unix – ZABBIX Get

Jde o pomocný program, který umí komunikovat s agentem a přijímat od něj zasílaná data. Používá se hlavně při odhalování příčin problému s agentem.

#### 5.4.6 Utilita pro Unix – ZABBIX Sender

Jde o program, pomocí kterého můžeme zasílat získaná monitorovaná data za ZABBIX server. Využívá se především pro dlouho běžící uživatelské skripty. Spouští se s parametrem konfiguračního souboru agenta a může zasílat pouze hodnoty, které jsou uvedené v sekci uživatelské parametry.

## 5.5 Webový server, administrace a správa

### 5.5.1 Webový server

Konfigurace jednotlivých objektů monitorování je přesunuta na webové rozhraní, které poskytuje přehlednou orientaci a správu nejen těchto prvků. Mezi další důležité komponenty webového serveru patří například průvodce instalací, nastavení přístupových práv, provádění auditu veškerého pohybu na webovém serveru. Web je rozdělen do pěti částí: monitoring, inventář, reporty, konfigurace a administrace.

### 5.5.2 Administrace

#### Autentizace

Systém ZABBIX umožňuje tři typy autentizací:

- Interní databáze ZABBIXu (hesla jsou ukládány ve formátu MD5)
- Ověření proti externímu LDAP serveru.
- Autentizace proti HTTP serveru (Apache server).

#### Uživatelé a skupiny uživatelů

V Systému ZABBIX existují tři typy uživatelů podle přednastavených oprávnění:

- Uživatel
- Administrátor
- Super administrátor

Systém skupin uživatelů se používá pro definování jednotlivých práv uživatelů. V základním nastavení je připraveno deset skupin, které obsahují práva dělená do tří oblastí – webové rozhraní, používání API, ladění systému.

#### Média

Jedná se o komunikační kanály, kterými budou doručovány upozornění. Základní média jsou: email, jabber, SMS pomocí GSM modemu a externí skript. Pro jednotlivé uživatele můžeme vymezit několik typů médií, která se budou využívat podle závažnosti upozornění a časové doby (dny v týdnu, hodiny).

## Skripty

Založení objektů, které budou reprezentovat externí skripty určené pro monitorování. Po instalaci ZABBIX serveru jsou předdefinovány dva externí skripty pro volání: ping, traceroute. Na skriptech se při jejich zakládání musí vymežit práva pro jednotlivé skupiny.

## Audit

Slouží k zobrazení detailního přehledu všech uživatelských aktivit, které byly provedeny přes webový server.

## Fronty

Používá se pro zobrazení souhrnného přehledu výkonu jednotlivých komponent monitorovacího systému.

## Upozornění

Přehled zaslaných upozornění jednotlivým uživatelům pomocí médií.

## Mapování

Díky tomuto systému je možné usnadnit vyhodnocování a reprezentaci dat v grafických komponentách a zasílání upozornění pomocí médií. Princip spočívá zejména v přiřazení zpřesňujících popisů k číselným hodnotám.

Příklad mapování pro stav služeb Operačního Systému Windows:

- 0 ⇒ Služba běží
- 1 ⇒ Služba je pozastavená
- 3 ⇒ Služba čeká na pozastavení
- 4 ⇒ Služba pokračuje v čekání
- 5 ⇒ Služba čeká na zastavení
- 6 ⇒ Služba je zastavená
- 7 ⇒ Služba je v nedefinovaném stavu
- 255 ⇒ Služba nebyla nalezena

### 5.5.3 Konfigurace

#### Šablony

System šablon usnadňuje a zpřehledňuje monitorování jednotlivých stanic. System ZABBIX zavádí přímou provázanost služeb, položek a pravidel vyhodnocování monitorování na základě typu stanice. Šablony obsahují na první úrovni aplikace (služby), které na stanici jsou spuštěny, na druhé úrovni konfigurační položky aplikací a na třetí úrovni pravidla vyhodnocení (triggery). Samostatnou úrovní jsou grafy, které využívají jako vstupní data položky aplikací a pravidla pro vyhodnocení stavu monitorované položky. Pro rychlejší úpravu a vytváření můžeme využít možnost importu a exportu ve formátu xml.

Příklady šablon:

- OS Windows
- JAVA
- SNMPv1
- Tomcat
- App MySQL
- Cisco 837

#### Stanice a skupiny stanic

Jedná se o definici síťových objektů, které budeme monitorovat. Jednotlivým objektům lze přiřazujeme stejné objekty jako šablonám.

#### Konfigurační položky

Konfigurační položky jsou jednotlivé parametry určené pro monitorování. Při definici nové položky je nutné vyplnit její povahu – o jaký typ se jedná a na základě tohoto typu ZABBIX automaticky nabídne další volby. Mezi další parametry patří frekvence monitorování, časová délka uchování získaných hodnot, mapování, přiřazení k aplikacím. Nejdůležitějším parametrem je tzv. klíč. Jedná se o unikátní označení konfigurační položky, které obsahuje systém získání její hodnoty (makro). System ZABBIX obsahuje stovky maker, která můžeme využít při vytváření či úpravě konfiguračních položek.

## **Události**

Skládají se z pravidel (triggerů), automatické registraci nových agentů, automatické vytváření konfiguračních položek, pravidel a grafů na základě povahy stanice (souborový systém, síťová rozhraní, SNMP OID). Konfigurační položky mají definovaná pravidla vyhodnocení monitorování jejich hodnot. Na základě pravidel se spouští akce či skupina akcí (operací) – zaslání upozornění pomocí médií na uživatele či skupinu uživatelů, spuštění vzdáleného příkazu pro proaktivní pokus o nápravu aktuálního stavu aplikace.

## **Obrazovky, prezentace, mapy**

Pro znázornění logických bloků z jednotlivých objektů můžeme využít objekty typu obrazovka. Prezentace jsou množiny obrazovek, které se mění v předem definovaném intervalu a vzbuzují tedy dojem slajdů. Obrazovka je v podstatě tabulka, která v jednotlivých buňkách může obsahovat:

- Grafy
- Mapy
- Vnořené obrazovky
- Informace o vyhodnocených pravidlech, stavu stanic

Mapy se využívají k zobrazení jednotlivých síťových prvků a definice závislosti mezi nimi.

## **IT Služby**

Jedná se o pohled na fungování systému z obchodního hlediska. V mnoha případech není důležité zobrazit fungování systému na nejnižší úroveň jednotlivých položek, ale znázornit dostupnosti jednotlivých služeb. Tyto reporty slouží k definici počítání jednotlivých SLA založených na vyhodnocených pravidlech monitorování určitých konfiguračních položek. Díky hierarchické struktuře těchto služeb jde odhalit slabá místa systému a zaměřit se na jejich zdokonalení.

## 5.6 Monitorování provozu

Monitorování je založeno na kontrole pravidel pro jednotlivé konfigurační položky a spouštění následných akcí.

### 5.6.1 Pasivní kontroly

Pasivní kontrolou se označuje jednoduchá žádost ZABBIX serveru nebo ZABBIX proxy o zaslání určitých dat. Jde například o: vytížení CPU, využití disku, paměti. Agent tento dotaz zpracuje a zašle požadovaná data zpět.

#### Komunikace

Server vytvoří TCP spojení s agentem. Agent obdrží seznam klíčů, pro které má provést pasivní kontrolu. Agent zašle serveru hlavičku, která obsahuje pouze získanou hodnotu monitorované položky. Server uzavře spojení.

### 5.6.2 Aktivní kontroly

Aktivní kontrola vyžaduje mnohem komplexnější zpracování na straně agenta. Agent na začátku obdrží seznam monitorovaných konfiguračních položek a provádí samostatně v předem definovaných intervalech kontroly. Výsledky těchto kontrol samovolně posílá na ZABBIX server nebo ZABBIX proxy.

#### Začátek komunikace

Agent naváže TCP spojení s druhou stranou a zašle žádost o seznam kontrol. Server musí zaslat seznam položek, který obsahuje pro každou položku klíč a zpoždění. Agent provede analýzu přijatých dat a uzavře spojení. Poté již nic nebrání agentu se započítáním aktivních kontrol a sbíráním požadovaných dat.

#### Odeslání výsledků na server

Agent naváže TCP spojení s druhou stranou a zašle seznam získaných hodnot doplněný o bližší specifikaci kontroly. Zpráva obsahuje kromě hodnoty kontroly: stanici, klíč a čas, kdy byla kontrola provedena a data získána. Server zašle odpověď, která obsahuje: počet úspěšně zpracovaných, počet nezpracovaných, celkový počet přijatých položek a uplynulý čas využitý ke zpracování celé zprávy. Agent vyhodnotí zaslaný status a uzavře spojení.

## 5.7 Přídavné hodnoty systému ZABBIX

### 5.7.1 Automatické zakládání stanic a služeb

ZABBIX umožňuje definování akcí, které se provedou při automatické nové stanice či služby, která odpovídá definovanému rozsahu monitorovaných portů a IP adres. Díky možnosti specifikace jednotlivých podmínek, které agent musí splňovat (například výskyt určitého slova v jeho názvu či proxy, která nám tento dotaz zasílá), je možné jednoduše provést příslušné operace. Můžeme si vybrat z následujících operací:

- Zaslání upozornění.
- Spuštění vzdáleného příkazu.
- Přidání stanice.
- Přidání do určité skupiny stanic.
- Nastavení stanice jako neaktivní.
- Přidání odkazu šablon na tuto stanici (nastavení konfiguračních položek, akcí, pravidel pro tuto stanici z konkrétních šablon).

Příklad využití v praxi:

- Automatické založení stanic na základě operačního systému  
Agenti pro jednotlivé operační systémy a šablony, určené těmto operačním systémům, používají odlišné parametry a metody monitorování.
- Automatické založení komponenty monitorovaného systému  
Informační systémy se skládají zpravidla z jednotlivých komponent, které plní určitou roli. Na základě definice jednotlivých komponent, služeb, které na nich běží, kontrol, které se mají provádět, můžeme jednoduše automaticky přidávat tyto komponenty do seznamu monitorovaných stanic.

### 5.7.2 Opakované oznamování a eskalace

ZABBIX obsahuje velice jednoduchý a efektivní systém pro eskalace a opakované oznamování. V závislosti na konfiguraci bude ZABBIX postupně stupňovat upozorňování na neřešení určitého problému. Konfigurace umožňuje definování jednotlivých akcí pro konkrétní úroveň eskalací (opakování výskytu porušení určitého pravidla).

Eskalace mají za úkol:

- Informovat uživatele o přetrvávání problému.
- Eskalovat problém nadřazeným uživatelům a skupinám uživatelů.
- Spouštět vzdálené příkazy.
- Zaslát zprávu všem zainteresovaným uživatelům a skupinám uživatelů.
- Pozastavit eskalaci u problémů, u kterých došlo k potvrzení (tzv. podmíněná eskalace).

Příklad eskalace problému s databází PostgreSQL (obrázek 5.2):

- V první řadě jsou upozorňování lokální administrátoři PostgreSQL databáze.
- Po určité době (1h), kdy stále přetrvává tento problém, je upozorněna skupina databázových administrátorů.
- Za další hodinu je proveden vzdálený příkaz (v tomto případě restart databáze).
- Pokud problém stále přetrvává a nikdo ho nepotvrdil, dojde k upozornění skupiny globálních databázových administrátorů.

Action operations					
<input type="checkbox"/>	Steps	Details	Period (sec)	Delay	Action
<input type="checkbox"/>	1 - 2	Send message to Group "CZ PostgreSQL admins"	Default	Immediately	<input type="button" value="Edit"/>
<input type="checkbox"/>	2	Send message to Group "CZ Database manager"	Default	01:00:00	<input type="button" value="Edit"/>
<input type="checkbox"/>	3	Run remote commands	Default	02:00:00	<input type="button" value="Edit"/>
<input type="checkbox"/>	4	Send message to Group "Global database manager"	Default	03:00:00	<input type="button" value="Edit"/>
				<input type="button" value="New"/>	<input type="button" value="Delete selected"/>

Obrázek 5.2 ZABBIX – akce opakovaného oznamování a eskalace.

## 5.8 Centralizace a distribuované monitorování aplikací

ZABBIX server je od začátku přizpůsoben pro distribuované monitorování. Obsahuje hierarchizaci jednotlivých serverů. Distribuované nastavení umožňuje založení až tisíc uzlů. Díky hierarchické struktuře můžeme rozdělit řešení tak, aby každý uzel měl dostatečný výkon pro monitorování přidělených stanic, služeb a položek. Speciální typy uzlů nám umožňují monitorovat i velmi vzdálené lokality se nestabilním připojením.

### 5.8.1 Hlavní server (Master node)

Webové rozhraní umožňuje konfiguraci a zobrazení stavu monitorování. Na tomto uzlu definujeme ostatní podřízené uzly, kterým posílá konfiguraci. Jako hlavní uzel je označen každý uzel, který má alespoň jednoho potomka.

### 5.8.2 Podřízené uzly (ZABBIX servery)

Jsou provázány s hlavním serverem. Jde o další plnohodnotné ZABBIX servery, které jsou zodpovědné za monitorování přidělených lokalit. Odesílají v intervalech změny konfigurace, historii dat a události na nadřazený uzel. Data jednotlivých položek zůstávají lokálně uložena a nezasílají se. V případě rekonfigurace (zrušení distribuovaného monitorování) pracují zcela nezávisle.

### 5.8.3 Podřízené uzly (ZABBIX proxy servery)

Jde o sběrnou stanici bez webového rozhraní, která zastupuje ZABBIX server na vzdálené lokalitě. Sbírá data a posílá je nadřazenému uzlu. Obsahuje databázi pro uložení výsledků monitorování přiřazených agentů.

### 5.8.4 Agenti

Mohou být připojeni k libovolnému uzlu monitorovacího systému. Provádí kontroly a předávají je nadřazenému uzlu. V případě nedostupnosti si data mohou chvilkově ponechávat. Jsou na platformě nezávislí.

## 6 PROAKTIVNÍ PREDIKOVÁNÍ

### 6.1 Co je proaktivní predikování

Predikce je předpovídání něčeho, co bude, většinou na základě minulých a současných informací. Proaktivní predikce je jednání na základě předpovědi. Systém se snaží zabránit nežádoucímu stavu, který by podle předpovědi nastal v blízké budoucnosti.

### 6.2 Jak funguje proaktivní predikování

Predikované parametry systému můžeme zpravidla popsat pomocí rovnic. Predikce se tímto krokem výrazně zjednoduší pouze na nalezení jejich řešení (následující hodnoty parametru). Některé parametry mohou být popsány velice složitými systémy rovnic, u jiných je nalezení takového popisu prakticky nemožné. Systém rovnic rovněž naráží na časovou závislost – čím složitější systém rovnic, tím je vyžadováno více operací k nalezení řešení a tedy roste i časová náročnost. V těchto případech se klasicky využívá neuronových sítí, které pracují na základě trénovacích množin k získání schopnosti predikce. Výhoda neuronových sítí spočívá v možnosti automatického zdokonalování.

### 6.3 Nástroje proaktivního predikování v dohledových systémech

Predikce byla v těchto systémech implementována pomocí systému rovnic, kde výsledek jejich řešení je predikovanou hodnotou měřeného prvku systému. Systém ZABBIX v další verzi bude využívat matematicko-statistickou Metodu nejmenších čtverců[7]. Tato metoda umožňuje nalezení aproximační funkce pro naměřené hodnoty prvku systému. Aproximační funkce je hledána jako lineární kombinace předem známých funkcí, ke kterým jsou vypočítány jednotlivé koeficienty. Zjednodušeně by se dalo říci, že je jedná o nalezení takové funkce, u které bude součet druhých mocnin chyb (rozdílů hodnot aproximační funkce a naměřených hodnot prvku systému) minimální. Na základě součtu druhých mocnin, tedy čtverců, byla tato metoda pojmenována. Standardně se používá aproximace pomocí lineární funkce, parabolou či polynomem.

## 7 TISKOVÁ ŘEŠENÍ

Tisková řešení vznikla na základě rozpínajícího se vývoje a používání tiskových zařízení, které nedokázaly poskytnout požadované funkčnosti. Tiskových zařízení je v dnešní době bezpočet. Jedná se hlavně o lokální tiskárny, síťové tiskárny, skenery, multifunkční zařízení, produkční tiskárny, velkoformátové tiskárny – plottery, 3D tiskárny a další. Výrobci tiskáren (vendoři) často vyvíjejí vlastní tisková řešení, popřípadě implementují nové funkčnosti přímo do tiskových zařízení. Každý kontinent má svá specifika a proto na nich najdeme několik tiskových řešení, která zde dominují. Kromě globálních tiskových řešení existuje nespočet menších lokálních řešení, která se většinou zaměřují na určité dílčí specifické funkce či integrace se systémy třetích stran.

Mezi nejznámější a nejrozšířenější tisková řešení současnosti patří:

- Equitrac [9]
- YSoft SafeQ [10]
- uniFLOW [umiFLOW]
- pcounter [12]
- SafeCom [13]
- MyQ [14]

### 7.1 Základní charakteristika

Některá tisková řešení jsou zaměřena pouze na určitý typ tiskových zařízení, jiná podporují širokou škálu vendorů. I přestože každý vendor má svá specifika, jsou tisková řešení postavena na stejných pilířích, který se dovíjí od základních vlastností tiskových zařízení. Mezi tyto pilíře pro tisk, kopírování, skenování a faxování patří:

- Zabezpečení
- Monitoring
- Správa práv uživatelů
- Účtování
- Statistický přehled

## 7.2 Obecného tiskového řešení

Budeme uvažovat systém, které by pokrýval základní pilíře tiskových řešení a nazvěme tento systém „Obecným tiskovým řešením.“ V následujících kapitolách se budeme věnovat tomu obecnému tiskovému řešení, vydefinujeme si základní charakteristické rysy pro jednotlivé funkce a možnosti jejich implementace s návazností na monitoring.

„U tiskových řešení rozhodně nelze říci, že by větší vždy znamenalo lepší. Často je tomu spíše naopak. To, že jste velký úřad nebo velká obec, nemá rozhodně přímou souvislost s tím, že si musíte pořídit velké tiskové řešení. Velká tisková řešení jsou určena pro velké objemy tisku, kde dokáží nabídnout suverénně nejlepší poměr cena/výkon. Ne každý velký úřad ale takováto zařízení potřebuje. Často je vhodnější třeba více menších zařízení s tím, že každá kancelář bude mít k dispozici své vlastní. Při výběru tak nehleďte na velikost úřadu, ale na to, jak velké tisky dané oddělení, odbor či část úřadu potřebuje.“ [15]

Zaměříme se na tisk, kopírování a skenování (faxování nebudeme uvažovat, protože v praxi u tiskových řešení jde pouze o jinou formu skenování).

### 7.2.1 Tisk

Na začátku musíme vytvořit tiskovou úlohu, kterou je možné vytisknout na tiskovém zařízení. V operačním systému přidáme nové zařízení, kterému přiřadíme specifický ovladač. Tento ovladač v sobě nese vlastnosti tiskárny – např. zda tiskárna je barevná či černobílá, kolik má tiskárna zásobníků, možnost nastavení sytosti tisku, konverze počtu stran na jednu stránku, atd. Tiskové ovladače obsahují zároveň i tiskový jazyk, pomocí kterého bude dokument konvertován do kódu, který tiskárna umí zpracovat.

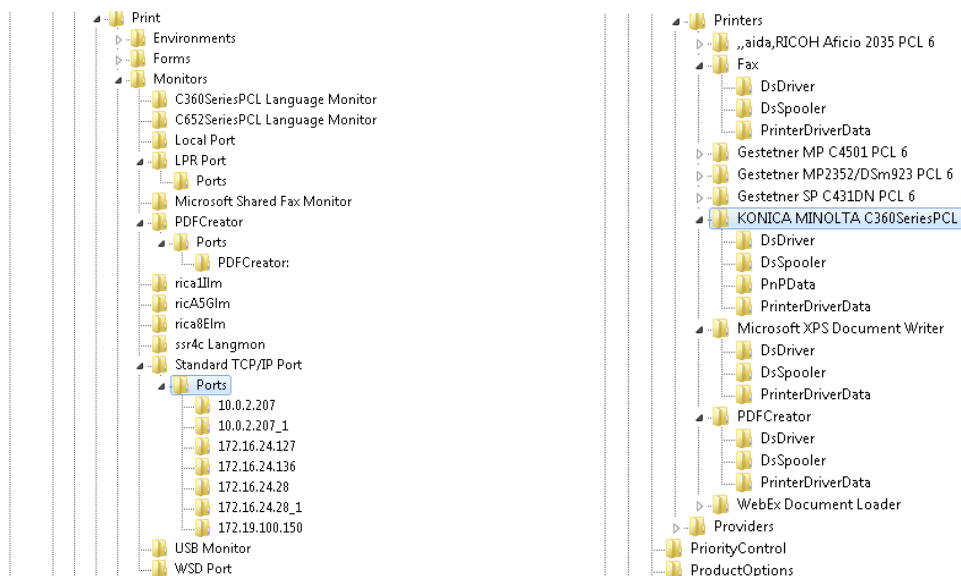
Mezi nejznámější a nejrozšířenější tiskové jazyky patří:

- PCL (Printer command Language)
- PS (PostScript Language)
- PJL (Peripheral Job Language)
- HPGL (Hewlett-Packard Graphics Language)

Po vytvoření tiskové úlohy je nutné ji transportovat na tiskárnu. K tomu se nabízí v operačních systémech několik TCP/IP protokolů, které se pro tiskárny přiřazují jako porty (tiskové monitory). Mezi nejznámější patří:

- LPR/LPD
- Raw TCP/IP
- HP JetDirect
- IPP

Po vytvoření a přiřazení portu k tiskárně se na systému Windows vytvoří nová položka v registrech, která obsahuje parametry tohoto portu. Tiskárna, která používá tento port, je také ukládána do registrů a obsahuje položku, která na používaný port odkazuje. Tento odkaz je založen pouze na názvu portu, proto systém vyžaduje unikátní název pro každý port. Při zakládání nového portu se stejnou IP adresou jsou automaticky doplněny do názvu portu znaky podtržítka a pořadového čísla pro zachování unikátnosti. Pro ukázkou použijeme port pro standardní TCP/IP porty – LPR, Raw a speciální port pro aplikaci PDF Creator, která slouží k zachycení tiskového dokumentu a transformování do formátu PDF. Výpis registrů můžete najít v příloze P II – P IV. Struktura registrů pro uložení tiskových portů a tiskáren v operačním systému Windows je zobrazena na obrázku 7.1. Tyto registry se nacházejí v `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control`.



Obrázek 7.1 Tiskový port, tiskárna – registry

Tiskové porty LPR a Raw obsahují společné parametry jako je např. ip adresa tiskárny, použitý port či SNMP komunita. Port pro aplikaci PDF Creator se výrazně od těchto protokolů liší. Obsahuje například parametr *Command*[Příloha P IV] jako definici cesty ke spustitelné aplikaci.

Při zasílání tiskové úlohy je úloha zařazena do tiskové fronty a posílána na cílovou IP adresu. Příjemce (tiskové zařízení či tiskový server) tuto úlohu přijme a zpracuje. Každý port dokáže odesílat najednou pouze jednu úlohu. Ostatní úlohy se řadí do tiskové fronty, která je postupně obsluhována tiskovým monitorem a odeslána na cílové místo. Operační systém Windows umožňuje sdružování monitorů pro jednotlivé tiskárny – takzvaný print pooling. Je tedy možné vytvořit více tiskových portů a tyto porty přiřadit pomocí print poolingu jedné či více tiskárnám. Operační systém bude pro vybírání tiskových úloh z fronty využívat vždy volný tiskový monitor. V praxi se může jednat například o použití jedné virtuální tiskárny v operačním systému pro distribuci tisků na více tiskáren. Nevýhodou je nutnost obejít všechna tisková zařízení a zjistit, na které fyzické tiskárně byl vlastně dokument vytištěn. Tisková zařízení nepodporují tisk více úloh najednou, proto vytvoření více tiskových portů se stejnou IP adresou by nemělo smysl.

Obecné tiskové řešení by mělo podporovat přijímání tiskových úloh a tedy implementovat základní tiskové protokoly pro příjem více úloh najednou. Tiskové úlohy se poté budou na tisková zařízení zasílat pomocí příslušného tiskového protokolu, který tato tiskárna podporuje (je u ní nastaven). Další možností implementace je práce s tisky na bázi tiskových front OS, kdy tisky necháme v čekající frontě a při požadavku na tisk je pouze přesune do vybrané přímé fronty, která za nás provede odeslání na tiskárnu.

Vzhledem k popsaným strukturám tiskových jazyků by obecně bylo možné zjistit z tiskové úlohy počet stran, barevné pokrytí a transformovat tiskovou úlohu z jednoho tiskového jazyku do jiného. Jednotliví vendori mají svá specifika a tedy nebude obecně možné vytvářet tiskové úlohy pomocí jednoho ovladače pro různé vendory. Problematiku specifických funkcí vyřešili někteří vendori vyvinutím tzv. univerzálních ovladačů – například: Universal Print Driver[16], Xerox Global Print Driver [17]. Tyto ovladače zajišťují tisk na většině tiskových zařízení pro konkrétního vendora, chybí ovšem některé funkce: například sešívání, tisk z bočního zásobníku, a jiné.

### 7.2.2 Kopírování

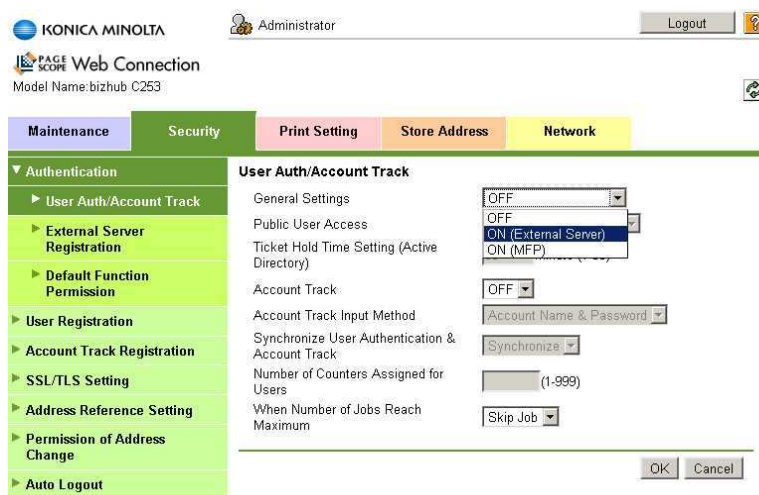
Kopírování se výstupem a průběhem neliší tolik od samotného tisku. Tiskové zařízení přijme data, tyto data si přeloží a převede je na výstup – vytištěná strana. Za kopírování bychom mohli označit i tisk z přenosného paměťového média, které většina tiskových zařízení v dnešní době podporuje (například USB Flash Disk). Pokud se zaměříme na první fázi, která je nejvíce odlišná od pouhého tisknutí úloh, zjistíme, že nám tu chybí důležitý prvek. Jedná se o identifikaci autora kopírovací úlohy. U tiskových úloh se jednalo o uživatele, který danou tiskovou úlohu v operačním systému vytvořil. Tento údaj byl obsažen přímo v tiskové úloze či byl vyplněn u vlastníka tiskové úlohy. U kopírování by bylo možné uživatele identifikovat z vlastníka souboru, který byl vytištěn z paměťového médi, které do frekvence není majoritní pro kopírování. Kopírování ve větší míře spočívá ve vložení vstupních dat ve formátu fyzického papíru, který byl předtím potištěn, na skenovací panel či do podavače, zvolení vybraných parametrů kopírování a použití tlačítka pro kopírování. Parametry kopírování jsou většinou totožné s parametry tisku, které lze vybrat v operačním systému.

Systém identifikace uživatele se tedy bude muset implementovat pomocí ověření uživatele pomocí externího zařízení, či pomocí panelu tiskového zařízení. S vývojem tiskových zařízení bylo ověření u některých tiskových zařízení přímo implementováno formou napojení na externí uživatelské systémy. Jedním z nejznámějších a nejrozšířenějších je napojení na uživatelský systém pomocí LDAP protokolu. Nastavení povolení používání LDAP protokolu je zobrazeno na obrázku 7.2. Detailní nastavení tohoto připojení zobrazeno v příloze P V.



Obrázek 7.2 LDAP pro tiskové zařízení Konica Minolta, převzato z [18].

Vývoj tiskových zařízení se nezastavil pouze na ověření uživatele, zda tento uživatel je zaveden v uživatelském systému a má určitá práva, ale byl doplněn i o monitorování činností uživatele. Systém ověření a monitorování činností uživatele byl implementován různými způsoby. Na jedné straně se jedná o ukládání těchto informací do paměti či na disk tiskového zařízení a na druhé straně jde o posílání těchto informací na externí server (tiskové řešení), který tyto informace zpracuje. Nastavení ověření a monitorování činností uživatele je zobrazeno na obrázku 7.3. Díky ověření uživatele je možné tiskovému zařízení předat informace o právech uživatele a omezit jeho činnost na tomto zařízení. Mezi nejpoužívanější omezení pro kopírování se řadí povolení a zakázání barvy. Tisková zařízení tento zákaz interpretují různě se stejným výsledkem – uživateli se podaří či nepodaří tiskárnu přimět použít barvu. Tento mechanismus je o to složitější u tiskáren, které obsahují speciální vlastnost – tzv. třívrstvé účtování. Jde o technologii účtování podle množství využití barvy na jednotlivých stránkách. Účtování se tedy nedělí pouze na černobílé a barevné, ale také na jednobarevné (monocolor). Nejedná se v tomto případě o fyzické použití pouze jedné barvy, jak by název mohl napovídat, ale o použití malého množství různých barev na stránku. Tiskárna před samotným tiskem kopírované strany vyhodnotí, kolik bude potřeba použít barvy a označí tuto stranu jako jednobarevnou kopii. V praxi jde o ušetření nákladů, neboť například testovací strana, kterou si můžete vytisknout na každém ovladači operačního systému Windows má malé logo tohoto systému barevné. Na normálních tiskárnách by tato strana byla označená za barevné kopírování a zaplatili byste za ni stejně jako za obrázek přes celou stranu.



Obrázek 7.3 Nastavení monitorování přihlášení a jednotlivých činností uživatele, převzato z [19].

### 7.2.3 Skenování

Skenování má v první fázi stejný postup jako kopírování. Uživatel se musí na tiskovém zařízení ověřit, poté provede oskenování připravených materiálů a tiskové zařízení je převede do určité formy. Tisková zařízení podporují různé výstupní formáty. Uživatel má na výběr například z těchto výstupních typů souborů:

- PDF, PDF, vícestránkové PDF
- JPEG \ JPG
- TIFF, MULTI TIFF
- BMP

Dále má uživatel na výběr, jakým způsobem se dokument v digitální podobě dostane na vybrané místo. Tisková zařízení podporují přenos souborů pomocí:

- FTP
- SMTP
- SMB
- Paměťové zařízení
- Aplikací včetně OCR
- HTTP

### 7.2.4 Faxování

Faxování můžeme označit za určitou formu skenování. Vytvoření dokumentu probíhá stejně jako u skenování. Odlišnost můžeme najít v poslední fázi – v transportu dokumentu. V případě, že tiskové zařízení nemá v sobě zabudovaný fax, je tato služba zajišťována pomocí standardního posílání naskenovaným dokumentem na příslušnou na příslušnou emailovou adresu či uložení do příslušného adresáře a externí software zajistí odeslání faxu. Tisková zařízení mohou podporovat: odeslání barevného faxu, reporty o faxech, přesměrování na jiný fax/email, PC a LAN Fax (pouze odesílání), zabezpečený příjem faxu, zabránění nevyžádaných faxů, příjem faxu do paměti, automatické zmenšení souboru.

## **II. PRAKTICKÁ ČÁST**

## **8 NÁVRH IMPLEMENTACE PRO TISKOVÉ ŘEŠENÍ YSOFT SAFEQ**

### **8.1 Požadavky na monitorovací systém**

YSoft SafeQ je distribuované tiskové řešení založené na integraci tiskového řadiče (spooleru). Vzhledem k velice komplexnímu a specifickému systému není možné jednoznačně vybrat monitorovací systém, který by splňoval všechny požadavky a je zapotřebí rozšířit některý ze stávajících systémů pro monitorování a správu.

#### **Monitorování služeb a stavů stanic**

Systém by měl být schopen v pravidelných intervalech kontrolovat dostupnost jednotlivých služeb, parametry operačního systému a stavy stanic.

#### **Robustnost a odolnost vůči síťovým výpadkům**

Systém by měl obsahovat prvky, které umožňují monitorovat služby a stanice, které jsou vzájemně propojeny nestálým síťovým připojením. V případě krátkodobých výpadků spojení s hlavním serverem musí být sbírány informace o stavu jednotlivých prvků. Systém by měl podporovat jednoduchou škálovatelnost.

#### **Monitoring a správa relační databáze PostgreSQL a Microsoft SQL Server**

Tyto relační databáze jsou úložištěm dat centrálních serverů a serverů pro ukládání statistik tiskového řešení. Nutností je získávání informací o stavu těchto databází, obsahu vybraných tabulek a možnost úpravy jejich konfigurace.

#### **Podpora monitoringu a správy aplikací na platformě JAVA**

Centrální server obsahuje webové rozhraní založené na Apache Tomcat. Centrální server i distribuované servery jsou implementované na platformě JAVA. Požadavkem je monitoring a správa běhu JAVA aplikací. Distribuované servery neobsahují databázi pro ukládání dat, ale využívají souborový systém a paměť.

### **Predikční systém a mechanismus proaktivního jednání**

Systém musí obsahovat predikční mechanismy na předejití výskytu chybového stavu. Systém by měl být schopen automaticky reagovat na základní problémy (restart neběžící služby, rekonfigurace služby – změna parametrů konfigurační parametrů v souborech či tabulkách relačních databází, spuštění, zastavení či spuštění částí služeb na platformě JAVA) a pokusit se vykonat nápravu před zasláním upozornění o výskytu problému.

### **Multifunkční systém zaslání upozornění a eskalace problému**

Zasílání upozornění by mělo probíhat pomocí tří kanálů v závislosti na závažnosti a době výskytu problému:

- komunikační server Jabber – méně závažné problémy s upozorněním na první skupinu uživatelů v průběhu pracovních hodin
- email – upozornění na méně závažné problémy první skupinu v době nepracovních hodin, eskalace méně závažných problémů na první skupinu v době pracovních hodin, upozornění na střední problémy druhou skupinu, eskalace závažných problémů na třetí skupinu
- SMS – eskalace středních problémů první skupinu, upozornění na závažné problémy první skupinu, eskalace závažných problémů na druhou skupinu

### **Ukládání historických dat a vytváření statistických grafů**

Historická data by měla být ukládána po dobu jednoho roku. Na základě naměřených dat budou moci vybraní uživatelé vytvářet statistické grafy a zobrazit si plnění zvolených SLA.

### **Webové rozhraní**

Webové rozhraní musí podporovat ověření uživatele pomocí loginu a hesla. Uživatelé budou moci na základě přidělených práv provádět tyto akce: zobrazení monitorovaných položek, vytváření statistických grafů, editace a přidávání monitorovaných objektů, vytváření a změny SLA, reakce na problém a jeho potvrzení, konfigurace monitorovacího systému.

## 8.2 Porovnání vybraných monitorovacích nástrojů

Systémy NAGIOS a ZABBIX jsem vybral na základě jejich vlastností. Oba tyto systémy jsou vyvíjeny již řadu let a patří do kategorie open source. Díky dlouhodobému rozvoji se staly nejrozšířenějšími a nejpoužívanějšími nástroji v oblasti monitorování a správy rozsáhlých informačních systémů. Monitorovací nástroje JConsole a VisualVM jsem zařadil jako rozšiřující nástroje pro správu a monitoring, neboť jsou úzce propojeny s programovacím jazykem JAVA, který byl použit pro vývoj tiskového řešení YSoft SafeQ.

### 8.2.1 ZABBIX

Komplexní monitorovací systém kladoucí důraz na jednoduchost správy a monitorování. Veškerá funkcionalita je již obsažena v ZABBIX Serveru a tedy zde není systém dohrávání přídatných rozšiřujících balíčků. Díky komponentám ZABBIX Proxy a ZABBIX Agent je možné monitorovat rozsáhlé informační systémy. Webové rozhraní poskytuje možnost konfigurace a vytváření komplexních výstupních sestav. Komponenty ZABBIX Server a ZABBIX Proxy využívají relační databáze (PostgreSQL, Oracle a jiné) pro ukládání konfigurace, výsledků monitorování a jejich synchronizaci mezi jednotlivými uzly. Automatickému přidávání nových ZABBIX Agentů a monitorovaných serverů na základě šablon jen podtrhuje propracovanou konfiguraci a správu tohoto řešení.

### 8.2.2 NAGIOS

NAGIOS obsahuje oproti ZABBIXu pouze monitorovací logiku, která spouští zásuvné moduly. Jádro systému se odproštuje od detailní znalosti monitorovaných prvků, kterou přenechává přídatným modulům. Díky systému přídatných modulů, kterých jsou nyní desítky, se stal komplexním řešením zaujímajícím jedno z prvních míst v oblíbenosti a používání. Díky svým komponentám NSCA klient a NAGIOS Agent je možné tento systém využít pro monitorování rozsáhlých informačních systémů. Integrovaný překladač Perlu umožňuje efektivnější využití zásuvných modulů napsaných v tomto programovacím jazyce. Neobsahuje propracované webové rozhraní jako ZABBIX.

### 8.2.3 JConsole a VisualVM

Tyto nástroje jsou určeny pro monitorování a správu aplikací běžících na platformě JAVA. Jejich hlavní funkcí je detailní zobrazení aktuálního stavu jednotlivých procesů a umožnění práce s objekty na základě architektury JMX. Oba tyto nástroje jsou standardní součástí instalačního balíčku JAVA SE. Díky rozšiřitelnosti o další moduly a variabilitu se tyto moduly staly důležitou součástí procesu monitorování JAVA aplikací. Tyto nástroje postrádají komplexnost pro možnost nasazení do rozsáhlých informačních systémů.

### 8.2.4 Výběr monitorovacího systému

Díky propracovanému webovému rozhraní poskytujícímu veškeré funkce pro konfiguraci, monitorování a správu informačních systémů, kompaktnějšímu vývoji, větší komunitě, integraci stovek monitorovacích funkcí a budoucímu predikčnímu systému byl vybrán komplexní systém ZABBIX. Nevýhodou tohoto systému oproti jeho konkurentovi NAGIOSu je absence funkce flappingu. Takto funkcionality se ovšem dá nahradit sofistikovanou volnou jednotlivých triggerů. Pomocné nástroje JConsole a VisualVM budou doporučeny v případě výskytu problému pro detailnější analýzu aktuálního stavu a získání důležitých dat pro vývojové oddělení.

## 8.3 Návrh implementace

Monitorovací systém zajišťující automatické dohledování a správu pro tiskové řešení obsahuje následující komponenty.

### 8.3.1 Centrální dohledový server

Centrální ZABBIX server tvoří jádro systému. Konfigurační parametry i vytvořené šablony pro jednotlivé typy komponent s přidělenými položkami a pravidly budou automaticky rozesílány podřízeným uzlům. Centrální server bude v rozložitějších podnikových systémech<sup>8</sup> použit pouze pro konfiguraci a sběr historických trendů a událostí. Pro ostatní podnikové systémy by měly být tyto servery instalovány vždy minimálně dva z důvodu překrytí případného výpadku jednoho serveru. Tyto servery budou fungovat zároveň pro vyhodnocování pravidel a provádění příslušných následných akcí při jejich porušení.

### 8.3.2 Podřízené uzly

Dva podřízené uzly budou instalovány na všech lokálních centrálách. Jejich úkolem bude komunikace s centrálním serverem – přijímání konfiguračních voleb, šablon s položkami a pravidly pro vyhodnocování, přijímání aktuálních hodnot monitorovaných položek od jednotlivých agentů a ZABBIX Proxy serverů. Lokální administrátoři budou mít zároveň možnost vytvoření dalších monitorovacích položek a pravidel, popřípadě změnu konfigurace, nezávisle na centrálním serveru. Tyto servery pracují i v době, kdy centrální server není dostupný a zajišťují správu a dohledování celé lokální oblasti. Podřízené uzly v menších podnikových systémech nemusí být instalovány, neboť jejich funkci převezmou centrální servery.

---

<sup>8</sup> Podnikové systémy, které mají strukturu minimálně tříúrovňovou. Z geografického hlediska se jedná o podnikové systémy s centrálou, hlavními pobočkami (lokální centrály) v několika zemích a lokálními menšími pobočkami, které jsou podřízené lokálním centrálám.

### 8.3.3 Lokální sběrná místa

Lokálním sběrným místem je ZABBIX Proxy server, který má za účel shromažďovat data od jednotlivých agentů, ukládat je do lokální databáze a v případě dostupnosti podřízených uzlů či centrálního serveru je odeslat k vyhodnocení. Tento server nevykonává aktivní monitorování. Jeho instalace je nutná na lokacích, které mají velice nestabilní síťové připojení k serveru, který vykonává vyhodnocení a spouštění příslušných následných událostí.

### 8.3.4 Monitorovací agenti

ZABBIX Agent bude instalován na všech serverech, které budou spravovány a dohledovány. Jeho úkolem bude provádět aktivní kontroly monitorovaných parametrů lokálních služeb a stanice. Vyhodnocené údaje budou odesílány na sběrná místa, podřízené uzly či centrální server. Tyto agenti budou fyzicky provádět následné akce v závislosti na vyhodnocených hodnotách monitorovaných položek. Agent musí být spuštěn pod uživatelem, který má dostatečná práva nejen na získávání monitorovaných položek, ale i na provádění těchto akcí – například restart služby, editace konfiguračních souborů, přístup do databáze. Pokud uživatel nemá dostatečná oprávnění, musí agent přistupovat k těmto položkám pod jiným účtem. Definice přístupového účtu bude součástí monitorovacího skriptu či skriptu volaného pro provedení následné akce. Používání více přístupových účtů není doporučováno, neboť při jejich centrální změně musí být změněny lokálně v používaných skriptech.

### 8.3.5 Webové rozhraní

Webové rozhraní bude dostupné na centrálním serveru a na podřízených serverech. Lokální administrátoři budou toto rozhraní využívat pro definování nových monitorovacích položek, pravidel, konfiguračních voleb, zakládání nových uživatelských účtů a skupin, reakci na výskyt problému a jeho řešení. Hlavní strana bude obsahovat souhrnný přehled stavů systému, stanic. Každý uživatel si může hlavní stranu upravit podle svých potřeb – například si přiřadit příslušné grafy, mapy či obrazovky mezi oblíbené a tím umožnit uložení jejich odkazu na svou hlavní obrazovku. Vyvolání zobrazení těchto komponent bude výrazně zjednodušeno. Použití hlavní obrazovky je znázorněno na obrázku 8.1. Díky rozdělení hlavní obrazovky do několika částí poskytuje administrátorům snadné

zorientování a přechod přímo k problému, který je indikován příslušným pravidlem. Pro rychlejší manipulaci s jednotlivými stanicemi bude sloužit systém šablon. Po přidání nové monitorované aplikace, položky, pravidla grafu či dalšího objektu do šablony je tento objekt automaticky přiřazen ke všem aktuálním stanicím, které mají spojení s touto šablonou.

PERSONAL DASHBOARD

Favourite graphs

- SafeQ\_CML1:CPU Loads
- SafeQ\_CML1:Disk usage

Graphs »

Favourite screens

- SafeQ\_CMLs
- Databases
- Print devices
- SafeQ\_CRS
- SafeQ\_ORSS
- SafeQ\_server - management
- SafeQ\_servers
- Zabbix\_server

Screens »

Favourite maps

- AT
- SK

Maps »

Status of Zabbix

Parameter	Value	Details
Zabbix server is running	Yes	127.0.0.1:10051
Number of hosts (monitored/not monitored/templates)	51	9 / 0 / 42
Number of items (monitored/disabled/not supported)	181	171 / 0 / 10
Number of triggers (enabled/disabled)[problem/unknown/ok]	68	68 / 0 [9 / 25 / 34]
Number of users (online)	2	1
Required server performance, new values per second	5.26	-

Updated: 23:32:35

System status

Host group	Disaster	High	Average	Warning	Information	Not classified
SafeQ_CML	0	0	8	0	0	0
SafeQ_CRS	0	0	0	0	0	0
SafeQ_ORSS	0	1	0	0	0	0
SafeQ_servers	0	0	8	0	0	0

Updated: 23:32:35

Host status

Host group	Without problems	With problems	Total
SafeQ_CML	2	1	3
SafeQ_CRS	1	0	1
SafeQ_ORSS	4	1	5
SafeQ_servers	2	1	3

Updated: 23:32:35

Last 20 issues

Host	Issue	Last change	Age	Ack	Actions
SafeQ_ORSS	SafeQ_ORSS is not reachable	06 May 2012 23:32:30	5s	No	-
SafeQ_CML1	FTP server is down on SafeQ_CML1	06 May 2012 23:04:26	28m 9s	No	-

Obrázek 8.1 ZABBIX server hlavní obrazovka

### 8.3.6 Databáze

Jako databáze bude doporučena relační databáze PostgreSQL[4] - díky licenčním podmínkám a robustnosti je lepším řešením než defaultní databáze MySQL. Pro rozsáhlejší podnikové systémy, které budou obsahovat stovky a tisíce monitorovaných položek, je tato databáze nutností. Nelze použít databázi MySQL. Databáze PostgreSQL je součástí tiskového řešení SafeQ, proto bude její instalace a správa pro administrátory jednodušší.

### 8.3.7 Aplikační logika

Sběr monitorovaných položek budou zajišťovat jednotliví agenti. Skripty, které rozšiřují stávající ZABBIX systém, budou implementovány v programovém jazyce JAVA, neboť je vyžadován na všech serverech tiskového řešení SafeQ. Pro ostatní tisková řešení je doporučeno použít vývojové nástroje, které jsou nutné pro běh toto systému. Nebude tedy zapotřebí instalovat dodatečně další software. K monitorování a správě databází PostgreSQL a Microsoft SQL Server budou použity nástroje příkazové řádky, které jsou již obsaženy v operačních systémech nebo jsou součástí instalace těchto databází. Jmenovitě jde o psql a osql. Tyto nástroje umožňují specifikaci uživatele, pod kterým jsou dotazy volány. Některé operace s databázemi mohou být vykonány pouze speciálními uživateli.

Komponenty tiskového řešení, které běží na platformě JAVA, budou obohaceny o spouštěcí parametry, které umožňují management (-Dcom.sun.management.jmxremote).

Mezi komponenty patří:

#### **Centrální servery (CML)**

Tyto servery obsahují webové rozhraní, databázi pro ukládání informací a aplikační logiku. Vzhledem k možnosti připojení aplikace na externí databázi a omezeného počtu těchto serverů pro jedno řešení nebudeme vytvářet speciální šablonu pro servery s interní a externí databází zvlášť. Vytvoříme dvě šablony pro komponenty CML a databázi. Pokud server obsahuje interní databázi, stanici v ZABBIXu přiřadíme také tuto šablonu. ZABBIX obsahuje hlavní monitorované parametry pro JAVU, Web server, Microsoft SQL Server. Jediné, co chybí, jsou speciální parametry, které je nutné do jednotlivých agentů doplnit. Pro PostgreSQL můžeme využít rozšiřující aplikace ZABBIXu Postbix nebo napsat jednotlivé příkazy ručně – například dotaz pro velikost databáze:

```
UserParameter=psql.db_size[*],psql -t -c "select pg_database_size('$1')"
```

Šablona bude dále rozšířena o monitorování běhu služeb serveru CML a skripty, které se budou spouštět jako reakce na chybové stavy monitorovaných prvků (restart služeb, rozšíření přidělené paměti, změna maximálního počtu spojení, změna parametrů v konfiguračních souborech, spuštění vypnutého procesu pomocí vzdáleného příkazu jmx).

### **Distribuované servery (ORS)**

Tyto servery neobsahují webové rozhraní ani databázi. Jde o klienty, kteří dokáží plnit funkce i v době výpadku spojení s hlavním serverem. Na těchto stanicích budou instalováni ZABBIX Agenti. Šablona pro tyto servery musí obsahovat rozšíření o monitorování spuštěných služeb a objektů v paměti. Jednotlivé funkce budou založeny na implementaci stromu MBean pro tuto aplikaci. Důležitými parametry jsou počty držených objektů a počty objektů čekajících k odeslání na ostatní servery. Využití operací, které objekty MBean podporují, můžeme jednoduše zjistit, zda důležitý proces běží, popřípadě vyvolat jeho spuštění. V současnosti nejsou tyto MBeany v aplikaci ORS zahrnuty a je nutné je implementovat pro rozšíření možností monitorování. Jde o jedinou možnou cestu, jak zpřístupnit důležité údaje dohledovému systému.

### **Server statistik (CRS)**

Tento server neobsahuje webové rozhraní. Jako databáze je použit Microsoft SQL Server. Šablona může využít stávající šablonu určenou právě pro monitorování tohoto serveru s důrazem na běh jednotlivých komponent této databáze. Další důležitou součástí je monitorování běhu služeb tiskové aplikace a případná rekonfigurace a restart služby.

### **Práce s log záznamy**

Všechny komponenty tiskového řešení využívají externí knihovnu log4j, která zajišťuje logiku práce s log soubory. Díky této knihovně je možné separovat záznamy logů tak, aby jejich obsah a rotace usnadňovaly práci nejen administrátorům, ale také systému pro dohledování a správu. Je možné postupovat několika způsoby: rozdělení podle jednotlivých procesů, rozdělení podle typu záznamu (INFO, WARN, ERROR), rotace po hodinách, dnech, týdnech.

### **Monitorování tiskových zařízení**

Tiskový systém SafeQ obsahuje možnost vytvoření monitorování tiskáren. Uživatel si může zvolit stav tiskového zařízení a událost, která má být po jeho detekci spuštěna. Díky této funkcionalitě nemusí být do systému pro dohledování a správu tato funkcionalita integrována. Tisková řešení, která tuto možnost nemají, mohou využít předdefinované šablony pro monitorování SNMP obsahující veškeré informace o tomto zařízení. Díky protokolu SNMP je možné nejen získávat hodnotu určených parametrů, ale je možné i hodnoty parametrů měnit. Jediný problém nastává, pokud tiskové zařízení tuto

komunikaci vypne. Tisková zařízení mají ve většině případů webové rozhraní, které umožňuje vzdálený restart zařízení. Vyvolání této akce je poměrně složité a je silně závislé na firmwaru tiskového zařízení. Restart tiskového zařízení bude v tomto případě ponechán na administrátorech.

### **Nastavení prahových hodnot**

Prahové hodnoty lze odvodit na základě detailní znalosti daného prostředí a aplikace. Vzhledem k velké variabilitě prostředí, kde můžeme monitorovat počty stanic v řádu jednotek a stovek, je nutné určit prahové hodnoty na základě empirického měření přímo na daném prostředí. Prahové hodnoty budou stanoveny po ukončení tohoto měření. Na základě několika měření bude možné stanovit povahu jednotlivých parametrů a určit jejich závislost na typu prostředí.

## **8.4 Možnosti rozšíření**

Některá z tiskových řešení podporují běh aplikací v programovacím jazyce JAVA. Jedná se například o tisková řešení RICOH, SAMSUNG a další. Na těchto zařízeních by bylo potenciálně možné s tiskovou aplikací spustit i její management. V tomto případě by bylo možné aplikaci spravovat a získávat informace, které nyní nejsou k dispozici. Dalším rozšířením. Tyto informace jsou ovšem chráněny a je tedy obtížné se k podrobnějším detailům propracovat. Tisková řešení, která integrují tyto tisková zařízení a umožňují instalovat tyto aplikace, mohou disponovat těmito informacemi a tedy vyvinout postup dalšího monitorování.

Dalším rozšířením bude bezesporu využití připravovaného predikčního systému, který systém ZABBIX bude v následujících verzích obsahovat. Predikční systém umožní identifikovat potenciální problém v době, kdy je možné tomuto incidentu zabránit. Nynější systémy umožňují pouze reagovat na incidenty, které již vznikly. Propracovaným systémem trendů získaných hodnot je možné se predikčnímu systému přiblížit. Cenou jsou ovšem hodiny strávené nad touto analýzou. Po zavedení predikčního systému bude možné operovat s hodnotami, které nastanou například v průběhu dalších tří kontrol, což může představovat časovou rezervu i v řádu desítek minut. Predikční systém bude možné ověřit pomocí zasílání upozornění bez vykonání preventivního zásahu, který by zabránil výskytu incidentu. Pokud incident opravdu nastane, jsou prahové hodnoty nastavené správně.

## ZÁVĚR

V práci autor shrnuje problematiku systémů pro dohledování a správu podnikových systémů s rozšířením o tiskový systém. K dané problematice dohledování přistupuje jak z teoretického, tak praktického pohledu. Studuje principy monitorování a monitorovací logiku. Diskutuje kvality vybraných systémů pro dohledování a jejich přínos pro distribuované systémy s rozšířením o tiskový systém.

Práce se zabývá komplexními systémy pro monitorování a správu. Představuje dva konkrétní monitorovací systémy spadající do oblasti volně šiřitelného softwaru – NAGIOS a ZABBIX v návaznosti na dohledování tiskových řešení. Srovnává jejich výhody i nevýhody pro dohledování vybraného tiskového řešení. Zdůvodňuje výběr dohledového systému spočívajícím v analýze požadavků kladených na tento typ dohledového systému. Práce zahrnuje analýzu obecného tiskového řešení na základě moderních tiskových řešení. Implementace dohledového systému s vybraným tiskovým řešením je provedena po prozkoumání architektury vybraného tiskového řešení a je úzce spjata s tímto řešením. Díky podobnosti tiskových řešení je možné využít poznatky získané touto prací pro jejich integraci se systémem dohledování a správy. V závěru autor diskutuje možnost rozšíření dohledovacího systému do oblasti monitorování integrovaných tiskových aplikací přímo na vybraných tiskových zařízeních, které představuje další krůček k plné integraci s tiskovými řešeními.

Největším přínosem této práce je podle autora vytvoření kostry monitorovací logiky a identifikování chybějících funkcionalit vybraného tiskového řešení. Získané poznatky mohou sloužit nejen pro rozšíření tiskového řešení ale i pro identifikaci slabín při jeho vývoji a testování.

## ZÁVĚR V ANGLIČTINĚ

This thesis summarizes problems of monitoring and management systems of enterprise systems with enhanced printing solution. The monitoring is discussed from both theoretical and practical point of view. Thesis contains studying the principles of monitoring logic, quality of monitoring and their benefits for distributed systems with enhanced printing solution.

The author chose system for monitoring and management from modern complex monitoring solutions. It represents two specific free complex monitoring systems – ZABBIX, NAGIOS. He compares the advantages and disadvantages of selected printing solution. The thesis includes analysis of the general printing solution based on modern printing solutions. Implementation of monitoring system with the selected printing solutions is made after analysing the architecture of the selected print solution. It is closely associated with this print solution. It is possible to choose and build monitoring system based on this thesis due to the similarity of printing solutions. In conclusion the author discusses the possibility of extending the system for monitoring and management of integration printing applications directly to the selected printing device. This represents a further step towards full integration with print solutions.

The greatest contribution of this work is creating a frame of monitoring logic and identifying the missing functionalities of the selected printing solution. The knowledge can be used to extend printing solution and to identify weaknesses in development and testing procedures.

**SEZNAM POUŽITÉ LITERATURY**

- [1] *NAGIOS* [online]. [cit. 2012-04-01]. Dostupné z:  
[http://nagios.sourceforge.net/docs/3\\_0/toc.html](http://nagios.sourceforge.net/docs/3_0/toc.html)
- [2] *ZABBIX* [online]. [cit. 2012-04-10]. Dostupné z:  
<http://www.ZABBIX.com/documentation/1.8/start>
- [3] Software architecture: 5th European Conference, ECSA 2011, Essen, Germany, September 13-16, 2011. proceedings [online]. 1st ed. New York: Springer, 2011 [cit. 2012-02-03]. ISBN 36-422-3797-5. Dostupné z:  
[http://books.google.cz/books?id=Z3ngxo-m3xoC&printsec=frontcover&hl=cs&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](http://books.google.cz/books?id=Z3ngxo-m3xoC&printsec=frontcover&hl=cs&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
- [4] PostgreSQL 8.4 [online]. 2009-07-01 [cit. 2012-04-12]. Dostupné z:  
<http://www.postgresql.org/docs/8.4/interactive/index.html>
- [5] PERRY, J. Steven. JAVA Management Extensions [online]. 1st ed. Cambridge [Mass.]: O'Reilly, c2002, 300 s. [cit. 2012-03-01]. ISBN 05-960-0245-9. Dostupné z: [http://books.google.cz/books?id=rLFkIKnCKGYC&printsec=frontcover&hl=cs&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](http://books.google.cz/books?id=rLFkIKnCKGYC&printsec=frontcover&hl=cs&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
- [6] *JMX Management* [online]. [cit. 2012-04-01]. Dostupné z:  
<http://docs.oracle.com/JAVAs6/docs/technotes/guides/management/>
- [7] DUCHOŇ, Bedřich. *Inženýrská ekonomika*. Vyd. 1. Praha: C. H. Beck, 2007, XIII, 288 s. ISBN 978-80-7179-763-0
- [8] Data Collection Protocols: SNMP, SMI, and MIB. *Etutorials.org* [online]. [cit. 2012-04-01]. Dostupné z:  
<http://etutorials.org/Networking/network+management/Part+I+Data+Collection+and+Methodology+Standards/Chapter+3.+Accounting+and+Performance+Standards+and+Definitions/Data+Collection+Protocols+SNMP+SMI+and+MIB/>
- [9] *Equitrac* [online]. [cit. 2012-04-10]. Dostupné z: <http://www.equitrac.com/>
- [10] *Y Soft Corporation a.s.* [online]. [cit. 2012-04-10]. Dostupné z:  
<http://www.ysoft.com/>
- [11] *Uniflow* [online]. [cit. 2012-04-10]. Dostupné z: <http://www.uniflow.eu/>

- [12] *Pcounter* [online]. [cit. 2012-04-10]. Dostupné z: <http://pcounter-europe.com/s>
- [13] *SafeCom* [online]. [cit. 2012-04-10]. Dostupné z: <http://www.safecom.eu/>
- [14] *MyQ* [online]. 2011 [cit. 2012-04-10]. Dostupné z: <http://www.myq-free.cz/>
- [15] JAK VYBÍRAT VELKÁ TISKOVÁ ŘEŠENÍ. In: *Parlament-vlada.eu: Informační technologie* [online]. 2011 [cit. 2012-04-10]. Dostupné z: <http://www.parlament-vlada.eu/index.php/hlavni-temata-is-ict/340-jak-vybirat-velka-tiskova-reeni>
- [16] Universal Printer Driver. *Konica Minolta US* [online]. [cit. 2012-04-15]. Dostupné z: <http://kmbs.konicaminolta.us/wps/portal/web/home/support/upd>
- [17] Global Print Driver. *Xerox* [online]. [cit. 2012-04-15]. Dostupné z: [http://download.support.xerox.com/pub/docs/GLOBALPRINTDRIVER/userdocs/any-os/en\\_GB/gpd\\_install\\_guide\\_en.pdf](http://download.support.xerox.com/pub/docs/GLOBALPRINTDRIVER/userdocs/any-os/en_GB/gpd_install_guide_en.pdf)
- [18] Global Print Driver. *Wiki Tufts* [online]. [cit. 2012-04-15]. Dostupné z: <https://wikis.uit.tufts.edu/confluence/display/exchange2010/Enable+LDAP+on+Konica+Minolta+Bizhub>
- [19] The Konica Minolta Guide to LDAP. *Ethos* [online]. [cit. 2012-04-15]. Dostupné z: <http://www.ethosimaging.com/documents/guides/Konica%20Minolta%20LDAP%20Guide%20V1.2.pdf>
- [20] GÁLA, Libor. *Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi, technologie informačních systémů, řízení a rozvoj podnikové informatiky*. 1. vyd. Praha: Grada, 2006, 482 s. ISBN 80-247-1278-4.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

API	Application Programming Interface
CML	Central Management Layer
CPU	Central Processing Unit
CRS	Central Reporting System
ePN	Embedded Perl Nagios
FTP	File Transfer Protocol
HDD	Hard Disk Drive
HTTP	Hypertext Transfer Protocol
IPP	Internet Printing Protocol
KM	Konica Minolta
LDAP	Lightweight Directory Access Protocol
LPD	Line Printer Daemon protocol
LPR	Line Printer Remote protocol.
MD5	Message-Digest Algorithm
NSCA	Nagios Service Check Adaptor
ORS	Offline Remote Spooler
PING	Packet InterNet Groper
POP3	Post Office Protocol
SLA	Service Level Agreement
SMTP	Simple Mail Transfer Protocol
TCP/IP	Transmission Control Protocol (TCP) and Internet Protocol (IP).

**SEZNAM OBRÁZKŮ**

Obrázek 1.1 SNMP protokol převzato z [3]. .....	15
Obrázek 1.2 Základní struktura MIB, převzato z [8]. .....	16
Obrázek 3.1 Architektura JMX, převzato z [5]. .....	21
Obrázek 3.2 VisualVM - nastavení MBean vláken. ....	25
Obrázek 3.3 VisualVM - zásuvný modul Visual GC. ....	25
Obrázek 4.1 NAGIOS klient pro operační systém Linux\Unix, převzato z [1]. .....	29
Obrázek 4.2 NAGIOS klient pro operační systém Windows, převzato z [1]. .....	29
Obrázek 4.3 Zásuvné moduly systému NAGIOS, převzato z [1]. .....	31
Obrázek 4.4 Volání zásuvných modulů, převzato z [1]. .....	32
Obrázek 4.5 Pasivní kontroly, převzato z [1]. .....	35
Obrázek 4.6 Síťová topologie závislosti stanic a služeb pro vyhodnocení stavu DOWN a UNREACHABLE, převzato z [1]. .....	37
Obrázek 4.7 Komunikace distribuovaného serveru s NSCA démonem, převzato z [1]. .....	43
Obrázek 5.1 ZABBIX Proxy, převzato z [2]. .....	47
Obrázek 5.2 ZABBIX – akce opakovaného oznamování a eskalace. ....	55
Obrázek 7.1 Tiskový port, tiskárna – registry .....	60
Obrázek 7.2 LDAP pro tiskové zařízení Konica Minolta, převzato z [18]. .....	62
Obrázek 7.3 Nastavení monitorování přihlášení a jednotlivých činností uživatele, převzato z [19]. .....	63
Obrázek 8.1 ZABBIX server hlavní obrazovka .....	72

**SEZNAM TABULEK**

Tabulka 3.1 Tabulka žádostí JSR pro JMX. ....	20
Tabulka 3.2 Příklad rozhraní MXBeans .....	23
Tabulka 4.1 Předběžný stav stanice. ....	36
Tabulka 4.2 Konečný stav stanice.....	36
Tabulka 4.3 Změny Hard a Soft stavů. ....	38
Tabulka 4.4 Operace flappingu .....	40

## SEZNAM PŘÍLOH

Příloha P I Distribuované monitorování systémem nagios převzato z [1]

Příloha P II TCP/IP LPR Print monitor

Příloha P III TCP/IP Raw Print monitor

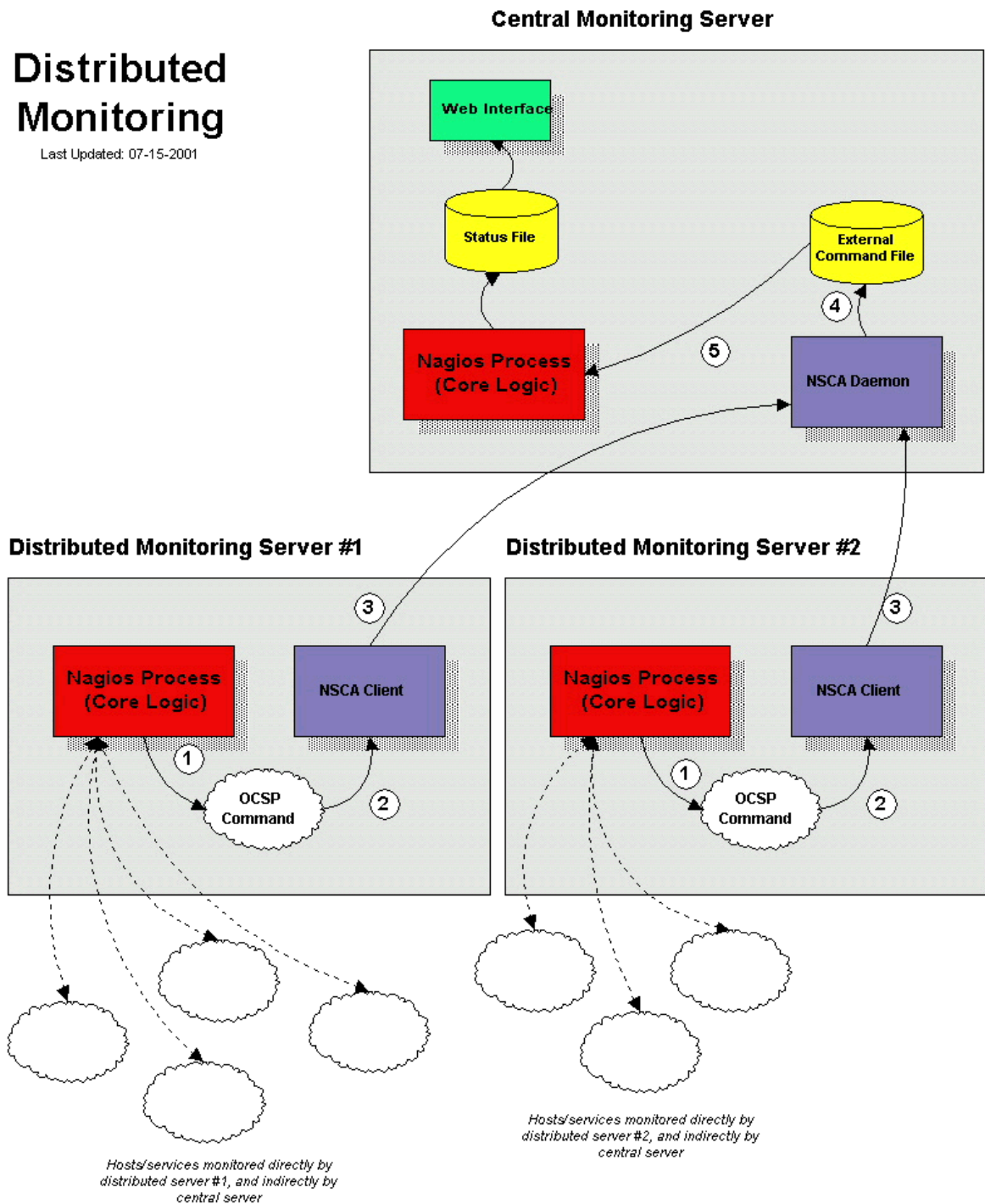
Příloha P IV PDF Creator print monitor

Příloha P V Nastavení připojení na AD pro tiskové zařízení Konica Minolta,  
Převzato z [18]

# PŘÍLOHA P I DISTRIBUOVANÉ MONITOROVÁNÍ SYSTÉMEM NAGIOS PŘEVZATO Z [1]

## Distributed Monitoring

Last Updated: 07-15-2001



## **PŘÍLOHA P II TCP/IP LPR PRINT MONITOR**

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Monitors\Standard TCP/IP Port\Ports\10.0.2.207]

"Protocol"=dword:00000002

"Version"=dword:00000002

"HostName"="10.0.2.207"

"IPAddress"=""

"HWAddress"=""

"PortNumber"=dword:00000203

"SNMP Community"="public"

"SNMP Enabled"=dword:00000001

"SNMP Index"=dword:00000001

"PortMonMibPortIndex"=dword:00000001

"Queue"="public"

"Double Spool"=dword:00000000

## **PŘÍLOHA P III TCP/IP RAW PRINT MONITOR**

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Monitors\Standard TCP/IP Port\Ports\10.0.2.207RAW]

"Protocol"=dword:00000001

"Version"=dword:00000002

"HostName"="10.0.2.207"

"IPAddress"=""

"HWAddress"=""

"PortNumber"=dword:0000238c

"SNMP Community"="public"

"SNMP Enabled"=dword:00000000

"SNMP Index"=dword:00000001

"PortMonMibPortIndex"=dword:00000000

## **PŘÍLOHA P IV PDF CREATOR PRINT MONITOR**

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Print\Monitors\PDFCreator\Ports\PDFCreator:]

"Arguments"="-PPDFCREATORPRINTER"

"Command"="C:\PROGRA~2\PDFCRE~1\PDFSpool.exe"

"Delay"=dword:0000012c

"Description"="PDFCreator Redirected Port"

"LogFileDebug"=dword:00000000

"LogFileUse"=dword:00000000

"Output"=dword:00000000

"Printer"="PDFCreator"

"Printerror"=dword:00000000

"Runuser"=dword:00000000

"ShowWindow"=dword:00000000

## PŘÍLOHA P V NASTAVENÍ PŘIPOJENÍ NA AD PRO TISKOVÉ ZAŘÍZENÍ KONICA MINOLTA, PŘEVZATO Z [18]

▶ TCP/IP Setting	<b>Setting Up LDAP</b>	
▶ E-mail Setting	No.	1
▼ LDAP Setting	LDAP Server Name	Tufts Directory
▶ LDAP Setting	Server Address	<input type="checkbox"/> Please check to enter host name. 130.64.3.21
▶ Setting Up LDAP	Port Number	389 (1-65535)
▶ IPP Setting	<input type="checkbox"/> Enable SSL	
▶ FTP Setting	Port Number (SSL)	636 (1-65535)
▶ SNMP Setting	Certificate Verification Level Settings	
▶ SMB Setting	Validity Period	Confirm
▶ Web Service Settings	CN	Do Not Confirm
▶ Bonjour Setting	Key Usage	Do Not Confirm
▶ NetWare Setting	Chain	Do Not Confirm
▶ AppleTalk Setting	Expiration Date Confirmation	Do Not Confirm
▶ WebDAV Settings	Search Base	dc=tufts,dc=edu
▶ OpenAPI Setting	Timeout	60 sec. (5-300)
▶ TCP Socket Setting	Max Search Results	100 (5-1000)
▶ IEEE802.1X Authentication Setting	Authentication Method	anonymous
▶ LLTD Setting	Login Name	anonymous
▶ SSDP Settings	<input type="checkbox"/> Password is changed.	
	Password	
	Domain Name	
	Select Server Authentication Method	Set Value
	Use Referral	ON
	Search Condition Attributes	Name
	Initial Setting for Search Details	
	Name	OR
	E-mail	OR
	Fax Number	OR
	Last Name	OR
	First Name	OR
	City	OR
	Organization	OR
	Organizational Unit	OR