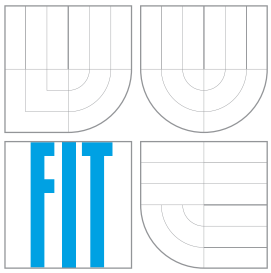
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MONITOROVÁNÍ SÍŤOVÉHO PROVOZU LINUXOVÉHO SERVERU

MONITORING NETWORK TRAFFIC OF LINUX SERVER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — PETR POSPÍCHAL
AUTHOR

VEDOUCÍ PRÁCE — Ing. JAROSLAV RÁB
SUPERVISOR

BRNO 2007

Zadání

1. Seznamte se operačním systémem linux. Zaměřte se na možnost zjištění informací o síťovém provozu, využití serveru uživateli. Seznamte se s protokolem a architekturou systému pro správu sítí pomocí SNMP.
2. Navrhněte model systému pro monitorování vybraných statistik. Statistiky systému budou poskytovány přes SNMP protokol a přes webové prostředí.
3. Implementujte navržený systém.
4. Proveďte testování systému.
5. Zhodnoťte dosažené výsledky a diskutujte další možnosti rozšíření systému.

Licenční smlouva

Abstrakt

V současnosti zažívá svět velmi rychlou expanzi Internetu. Stále více firem spoléhá na služby poskytované prostřednictvím síťových spojů mezi počítačovými systémy. S enormním nárůstem objemu přenášených dat vzrůstají i požadavky na systémy, narůstá i jejich složitost a je stále obtížnější udržet v chodu celý strom síťových zařízení. Klíčovým prvkem zajištění dostupnosti služeb, potažmo výdělků, je dokonalý přehled o funkčnosti jednotlivých prvků systému. Ve světě, kde většina centrálních uzlů Internetu funguje na operačních systémech typu UNIX je velmi vhodné mít nástroj, jenž umožňuje snadno přistupovat k informacím o stavu systému. Návrhem a implementací takového systému se zabývá tato bakalářská práce.

Klíčová slova

počítačová síť, síťový provoz, snmp, linux, server, monitorování

Abstract

Today the world is facing a very fast expansion of the Internet. Every day, more and more companies rely on services provided by network connections between computer systems. Enormous growth of data volume being transferred also requires more sophisticated systems. Thus it is more complicated to maintain the whole tree of network systems. Key issue to maintain reliability of services and profit, is to have perfect overview of each entity's availability. In a world where majority of important systems are being run on UNIX-type operation systems it is good to have a tool which allows easy access to information related to a system. The topic of this Bachelor work is to describe both concept and implementation of such a tool.

Keywords

computer network, network traffic, snmp, linux, server, monitoring

Citace

Petr Pospíchal: Monitorování síťového provozu linuxového serveru, bakalářská práce, Brno, FIT VUT v Brně, 2007

Monitorování síťového provozu linuxového serveru

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Rába

.....
Petr Pospíchal
10. května 2007

Poděkování

Tímto bych rád poděkoval vedoucímu mé bakalářské práce, Ing. Jaroslavu Rábovi, za jeho ochotně poskytnutý čas při konzultacích, motivaci a inspiraci. Setkání s ním pro mě byly přínosem do života.

Obsah

Kapitola 1

Úvod

Není tomu ani tak dávno, kdy ARPANET položila základ dnešnímu Internetu. Přestože od vzniku této sítě neuplynulo ani čtyřicet let, expanze jejího nástupce byla tak rychlá, že za tuto dobu stačila pokrýt doslova celý svět. Internet je dnes součástí asi většiny euroamerických domácností, znalost využívání jeho základních služeb se dnes považuje za samozřejmost. Stále větší množství lidí se s tímto médiem setkává denně, je na něj poukazováno ve stále větším množství reklam a mnoho zástupců mladé generace si bez něj neumí představit svůj běžný den. A co víc – stále více firem se nadobro zbavilo svých kamenných poboček a zvolili si raději jako obchod internet. Pro zákazníka je taková firma vlastně jenom textem a obrázky na obrazovce jeho počítače.

S každou novou generací produktů firmy Microsoft se práce pro uživatele jeví jednodušší, je stále více odstíněn od funkcionality softwaru, od jeho součástí a většinou i pokročilejších nastavení. Je pak snazší používat počítač, každý zběžně seznámený uživatel může posílat elektronickou poštu, hledat na webových stránkách informace, či si povídat se svým kamarádem přes jeho oblíbený instant messanger. V důsledku toho se internetová gramotnost a využívání internetu šíří ještě více. Chce-li firma získat pravidelné návštěvníky svých stránek a tím pádem i vydělat na reklamě, musí přijít na trh s něčím originálním. A tak se prosazují nové služby jako streamované video, internetová telefonie a pod. Úměrně s tím stoupá i využívaná kapacita síťových připojení, zvyšují se požadavky na servery poskytující služby a prvky v síti se stávají komplikovanější. S přibývajícím množstvím uživatelů a s tím spojené anonymity na internetu přibývá i online kriminality.

Pro řadu firem se internet stal jediným zdrojem příjmů, a tak na spolehlivosti služby, kterou poskytují, závisí jejich existence. Je tedy přirozené, že se snaží nějakou formou pojistit svoji živnost.

Klíčovým bodem pro zajištění funkčnosti jakéhokoliv systému je bez pochyby nutnost mít o něm stále dostatek informací. Pro tyto účely byl vyvinut protokol SNMP, který umožňuje jednotným způsobem získávat informace o síťových zařízeních. Tato data mohou být následně použita k vyhodnocení stavu systému, popř. jejich kolekce může sloužit k dlouhodobějšímu přehledu o monitorovaném prvku sítě.

Tato práce popisuje návrh a implementaci systému, který získává informace o svém hostiteli a poskytuje je přes protokol SNMP a webové rozhraní. Dává tak přehled o tom, v jakém je systém stavu, a to jak z hlediska dlouhodobého, tak i krátkodobého. Systém je určen pro operační systémy typu UNIX, které jsou hojně využívány jako servery poskytující služby na internetu.

1.1 Členění práce

Celá práce je členěna do kapitol s názvem podle tématického okruhu, které popisují.

1. Úvod

   kapitola 1, kterou právě čtete, slouží k uvedení do širšího kontextu problematiky. Nastiňuje také členění práce a návaznost na Semestrální projekt, který předcházel tvorbě této práce.

2. Požadavky na systém

   Následující kapitola, s číslem 2, se zabývá konkretizací pohledu na požadavky definované v zadání a sumarizací vlastností, které jsou nutné pro korektní návrh implementovaného systému.

3. Návrh systému

   Kapitola číslo 3 má za úkol čtenáři podrobně popsat návrh celého systému včetně jeho elementů. Je zároveň výchozím bodem pro následující kapitolu.

4. Implementace a testování systému

   Implementace a testování systému je kapitolou číslo 4, a popisuje problémy vzniklé při samotné implementaci, rozdíly oproti prvotnímu návrhu a také testování korektnosti výsledné implementace.

5. Závěr

   Poslední kapitola, s číslem 5, analyzuje dosažené výsledky, shrnuje vlastnosti systému a nastiňuje možnosti dalšího vývoje.

1.2 Návaznost na Semestrální projekt

V Semestrálním projektu byl prezentován především koncept funkcionality, který se v této zprávě vyskytuje v kapitole 3.1 na straně 7. Dále byla zmíněna volba statistik na základě SNMP stromu (kapitola 3.2 na straně 12) a v neposlední řadě také volba implementačního prostředí (kapitola 2.3 na straně 6). Většina popisu u kapitol byla kvůli omezenému času obhajoby Semestrálního projektu zkrácena, tudíž se v plnohodnotné formě vyskytuje až v této práci.

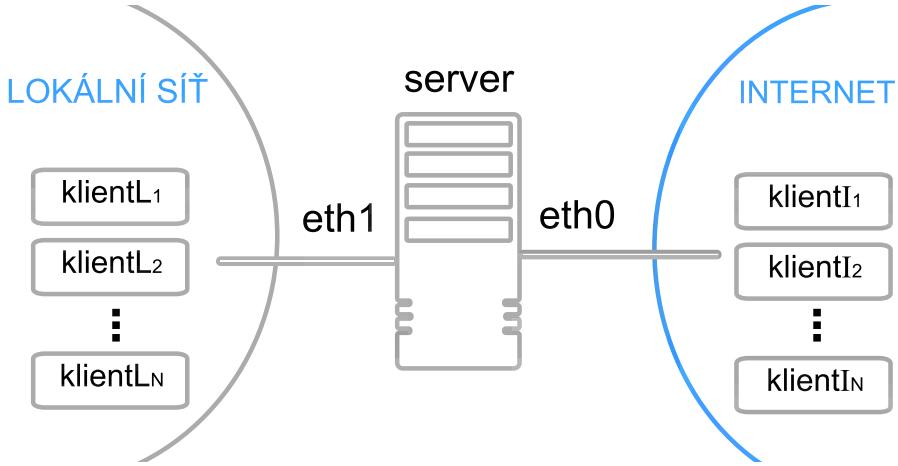
Kapitola 2

Požadavky na systém

Tato kapitola se zabývá konkretizací pohledu na požadavky definované v zadání a sumarizací vlastností, které by měl navrhovaný systém mít.

2.1 Testovací systém

Pro testování práce je třeba mít k dispozici vhodné prostředí, kde je možné v plné míře otestovat implementaci celého projektu. Bez pochyb je potřeba ověřovat implementaci v prostředí, ve kterém jsou poskytovány síťové služby a na kterém je operační systém typu Linux. Je vhodné, aby byl systém během testování průběžně vytěžován požadavky na tyto síťové služby, a tak byl sběr dat pro statistické zpracování (dále jen sběr statistik) prováděn nad dostatečně velkým vzorkem dat.



Obrázek 2.1: Zapojení testovacího systému

Jako testovací prostředí jsem si vybral server běžící na operačním systému Linux, konkrétně distribuci Gentoo. Jak je patrné na obrázku 2.1, tento server poskytuje služby jak pro lokální síť, tak pro internet. Přehled poskytovaných služeb je uveden v tabulce 2.1. Pro účely testování instalace byl stejný systém nasimulován i v prostředí virtuálního stroje (*Virtual Machine, VM*), jak popisuje kapitola 4.5 na straně 29.

Jak je z tabulky 2.1 patrné, pro klienty lokální sítě jsou poskytovány služby prostřednictvím

umístění klientů	název služby	název procesu	protokol aplikační vrstvy
lokální síť	fileserver	`samba`	CIFS
	překlad adres	jádro OS	-
internet	WWW	`apache`	HTTP
	databáze	`mysql`	-
	FTP	`proftpd`	FTP

Tabulka 2.1: přehled poskytovaných služeb na testovaném serveru

procesu `samba`, s využitím protokolu CIFS[1]. Naproti tomu `apache`, `mysql` a `proftpd` se starají o služby primárně pro klienty z internetu. Přímo jádrem operačního systému je pak zajištěn překlad adres (*Network Address Translation*-NAT), který slouží k přístupu klientů z lokální sítě do internetu.

Proces databáze je podstatný pro funkčnost systému, ale jeho provoz není zajímavý z hlediska monitorování síťového provozu.

Všechny výše zmiňované procesy beží na pozadí a svoje případná hlášení o stavu směrují do logovacích souborů. Ty jsou přístupné administrátorovi systému a je možné je s výhodou využít k diagnostice stavu, určení zatížení apod.

Detailnějším popisem získávání monitorovaných dat se zabývá kapitola 4.2 na straně 19.

2.2 Požadavky na statistiky

Účelem práce je, jak je zmíněno v kapitole 1, monitorovat hostitelský UNIX-ový systém z hlediska síťového provozu. Statistiky je proto vhodné vybírat tak, aby co nejvíce vypovídaly o stavu a dostupnosti síťových služeb, které monitorovaný systém poskytuje. Z hlediska toho je vhodné zabývat se především celkovou dostupností služby, množstvím přeneseným dat a počtem požadavků na obsluhu klienta. O celkovém stavu monitorovaného systému vypovídá i využití procesoru. Samotný sběr dat na hostitelském systému by neměl příliš ovlivňovat jeho chod.

Konkrétním výběrem statistik se zabývá kapitola 3.2 na straně 12.

2.3 Volba implementačního prostředí

Implementovaná práce by měla být snadno rozšiřitelná o nové statistiky a umožňovat tak adaptaci na nově poskytované služby.

Na platformě typu UNIX je hojně využíván modulární systém, kdy je finální funkcionalita dosažena spojením několika na sobě nezávislých programů, z nichž každý se specializuje na jednu konkrétní činnost. Pro tuto práci se podobný přístup jeví jako vhodný, protože by využíval prostředků hostitelské platformy a byl tak dostatečně malý, snadno rozšiřitelný a dobře konfigurovatelný.

Další detaily jsou k nalezení v kapitolách 3.1 a 4 na straně 7 resp. 17.

[1] Původní vývoj protokolu SMB *Server Message Block* byl zahájen Barry Feigenbaumem ve firmě IBM, později byl jeho vývoj převzat firmou Microsoft a spolu s uvedením nových funkcí byl v roce 1996 přejmenován na *Common Internet File System*. Je využíván pro účely sdílení prostředků (souborů, tiskáren a pod.) mezi počítači.

Kapitola 3

Návrh systému

Tato kapitola má za úkol čtenáři blíže popsat návrh systému jako celku, ale i jeho jednotlivých částí.

3.1 Koncept funkcionality

3.1.1 Způsob realizace

V kapitole 2.3 na straně 6 byla nastíněna možná realizace monitorovacího systému jakožto kompozice využívající standardní prostředky operačního systému. Tento způsob ralizace má několik výhod:

- Systémové prostředky jsou dobře odladěné a snadno použitelné
- Část prostředků je zahrnuta v operačním systému, a proto může být výsledný projekt menší
- Vzhledem k využití převážně skriptovacích jazyků je výsledný systém dobře modifikovatelný a snadno laditelný

mezi nevýhody pak patří především:

- Prostředky operačního systému nemusí být zpětně kompatibilní, nebo se mohou v nové verzi syntakticky či sématicky lišit
- Systém realizovaný prostředky operačního systému má omezenou přenositelnost

Vzhledem k tomu, že u projektu je kladen důraz především snadnou modifikovatelnost a potenciální využitelné prostředky jsou téměř shodné na celé platformě typu UNIX, jeví se realizace projektu s použitím prostředků operačního systému jako výhodná.

Jednou z hlavních předností operačních systémů typu UNIX je bezesporu skriptovací jazyk zabudovaný přímo do příkazového řádku. Mezi nejpoužívanější patří BASH (*Bourne-Again SHell*) [2], který světlo světa poprvé spatřil v roce 1978 a dodnes je oblíbeným nástrojem mnoha systémových administrátorů. Spolu se základními utilitami operačního systému tvoří velmi silný nástroj pro implementaci širokého spektra malých, ale i větších aplikací.

3.1.2 Ukládání dat

Jak je nastíněno v kapitole 2 na straně 5, je vhodné, aby samotné monitorování hostitelský systém příliš nezatěžovalo, protože primárním cílem serveru je poskytovat síťové služby a nikoliv monitorovat sám sebe. Zároveň je ale potřeba, aby projekt poskytoval údaje vypovídající o stavu v dlouhodobějším časovém měřítku. Proto je nutné, aby byly údaje o monitorovaných datech průběžně ukládány, aby bylo možné je později zpracovat například do podoby grafu, nebo je použít pro výpočet matematických statistických funkcí jako je směrodatná odchylka a pod. Periodický sběr dat přibližuje následující kapitola 3.1.3 na straně 8.

Pro účely ukládání dat se dá využít buď prostý soubor, nebo specializovaná aplikace – databáze. Mezi výhody ukládání dat do souboru patří:

- Data jsou uložena v člověkem čitelné formě, což umožňuje snazší ladění
- Přístup k souborům patří mezi základní možnosti většiny OS, není tedy potřeba žádné speciální softwarové vybavení
- Nízká paměťová náročnost

mezi nevýhody pak:

- Při větším objemu dat se nepříjemně projevuje lineární složitost přístupu k datům
- Nízká odolnost vůči poškození dat
- Omezené možnosti paralelního přístupu k datům

Protože mezi základní požadavky na systém patří, aby byly statistiky přístupné více uživatelům, jsou omezené možnosti paralelního přístupu značnou nevýhodou. Lze také očekávat, že za delší dobu bude objem nasbíraných statistik značný, a proto by lineární složitost přístupu k datům byla značnou překážkou pro plynulý chod systému. Vzhledem k tomu, že databáze patří mezi běžně poskytované služby serveru, není ve většině případů nutná její dodatečná instalace a paměťová náročnost se s nově přístupnými daty nijak dramaticky nezvyšuje. Přístup k datům je u ní navíc mnohem efektivnější, než u záznamů uložených v souboru.

Z výše uváděných důvodů je patrné, že pro ukládání dat je vhodnější zvolit databázi. Na systémech typu UNIX převažuje využití OpenSource [1] implementace databáze MySQL [6] (My-*Structured Query Language*, implementace strukturovaného dotazovacího jazyka), která plně postačuje pro použití v tomto projektu.

3.1.3 Periodický sběr dat

Jak již bylo nastíněno v kapitole 2 na straně 5, měl by implementovaný projekt poskytovat přehled o systému jak z hlediska období krátkodobého, tak dlouhodobějšího. Pro přehled o dlouhodobějším stavu musí být aktuální data periodicky ukládána do databáze, aby později mohla být použita pro statistické vyhodnocení.

[1] Mezi OpenSource se řadí takové projekty, u kterých je spolu s vlastním programem distribuován i zdrojový kód. Většina těchto projektů využívá licenci GPL (*General Public Licence*), která umožňuje volné upravování a distribuci takto modifikovaných projektů.

Periodické spouštění části systémů, která má na starost sběr dat, lze zajistit několika způsoby:

1. Aktivním čekáním, například cyklem s podmínkou v programu
2. Voláním systémové funkce `sleep`
3. Specializovanou aplikací, např. CRONd

První možnost, aktivní čekání, je nevhodným prostředkem, protože vytěžuje hadrware hostitelské platformy.

Druhá možnost, volání systémové funkce `sleep` [12], je lepší alternativou, nicméně u ní nastává problém s opětovným spuštěním programu po restartu systému, pádu programu a pod. Také by bylo potřeba dodatečně ošetřit dobu prvního spuštění, aby byly jednotlivé kolekce statistik sbírány například vždy na začátku minuty.

Poslední možnost, použití specializované aplikace, je vhodná, protože řeší výše zmiňované problémy. Navíc umožňuje automaticky hlásit administrátorovi systému výsledek operace sběru dat a snadno tak ladit chod systému.

Pro plánované spouštění aplikace na platformě UNIX je vhodný například CRONd [1], který se dá snadno nakonfigurovat na spoustění příkazu podle minuty, hodiny, dne v měsíci, měsíce, dne v týdnu, nebo kombinace předchozího. Časový interval pro sběr dat by měl být přiměřený, aby zbytečně častý sběr statistik příliž nevytěžoval systém a zároveň aby byly data v databázi dostatečně aktuální.

3.1.4 Webové rozhraní

Implementovaný projekt má mimojiné poskytovat informace skrze webové rozhraní. To je, narozdíl od protokolu SNMP, dobře uzpůsobené k poskytování graficky zpracovaných statistik vypovídajících o dlouhodobějším stavu systému.

Služby založené na protokolu HTTP (*Hyper Text Transport Protocol*, bezstavový protokol původně určený pro přenos HTML stránek) jsou dnes hojně rozšířené a jsou poskytovány na mnoha serverech. Díky tomu jsou aplikace pro tyto služby dobře odladěné a kvalitně zdokumentované. Mezi nejpoužívanější aplikaci pro poskytování WWW (*World Wide Web*) služeb patří webserver Apache [9], který je zářným příkladem výše zmiňovaných výhod hojného nasazení. Navíc je dobře konfigurovatelný a jeho funkčnost je rozšiřitelná skrze softwarové moduly. Je primárně určen pro nasazení na platformě typu UNIX, a proto se jeví jako velmi vhodný k použití v implementovaném projektu.

Stále se měnící data lze jen obtížně prezentovat na statických HTML (*HyperText Markup Language*) stránkách, a proto je třeba využít některého existujícího rozšíření, které zajistí dynamické generování stránek na straně webserveru. Pro webserver Apache je nejpoužívanějším takovým rozšířením softwarový modul `mod_php`. Využívá vlastní interpretovaný programovací jazyk PHP [8] syntakticky vycházející z jazyka C. Je dobře uzpůsobený pro práci s řetězci a databází, což z něj dělá dobrého kandidáta pro využití v tomto projektu.

Pro lepší přehled o dlouhodobém stavu systému je vhodné vizualizovat data do podoby grafu. K tomuto účelu lze použít buď externí aplikaci ve formě spustitelného souboru (např. GnuPlot), nebo některou z existujících knihoven pro skriptovací jazyk PHP.

Mezi výhody použití externí aplikace patří:

- Rychlost zpracování dat
- Velké množství podporovaných výstupních formátů

nevýhodami jsou:

- Nutnost instalace dodatečného softwarového vybavení
- Vyžaduje spouštění externí aplikace, což je pro webovou službu nevhodné

Z výše uvedeného shrnutí výhod a nevýhod je patrné, že použití externí aplikace pro generování grafů by mohlo mít pozitivní dopad na výkon systému. Aplikace pro generování grafů nicméně nebývají běžnou součástí vybavení serverů, a proto se jako vhodnější jeví použít existující PHP knihovnu, která může být snadno distribuována spolu s projektem.

Pro samotné generování grafů lze využít například knihovnu `JpGraph` [7], která generuje velmi kvalitní výstupy a přitom je snadno použitelná.

3.1.5 Poskytování dat přes protokol SNMP

Finální podoba projektu poskytuje statistiky jak přes webové rozhraní, tak přes protokol SNMP. Zatímco návrhem webového rozhraní se zabývá předchozí kapitola 3.1.4, SNMP je předmětem této kapitoly.

Simple Network Management Protocol, jak zní jeho plný název, je určen pro správu síťových zařízení. Za dobu své existence se rychle rozšířil a dnes je hojně využíván po celém světě. Díky tomu vznikla řada jeho implementací, které jsou nyní v pokročilých fázích vývoje. Mnoho z těchto implementací je dostupných jako OpenSource, a proto je vhodnější využít jich, raději než implementovat nový systém.

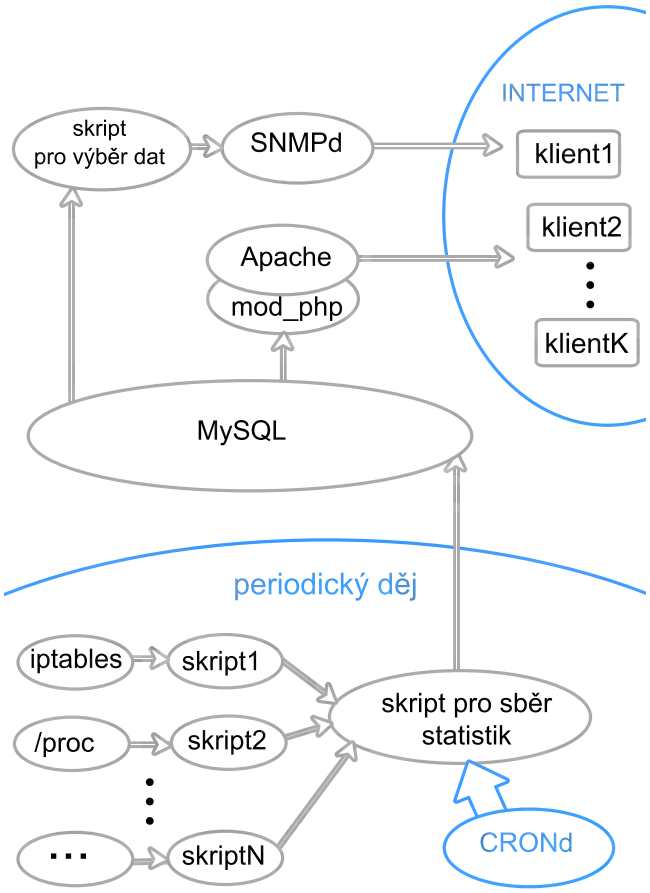
Kvalitním balíkem pro platformu UNIX poskytujícím celé spektrum aplikací spojených s protokolem SNMP je `NET-SNMP` [5]. Pro potřeby správy sítě jsou v něm k dispozici jak aplikace určené pro nasazení na monitorovaném systému (*network element*), tak aplikace pro vzdálené ovládání tohoto systému (*management station*).

Aplikací zodpovědnou za poskytování monitorovaných dat a ovládání hostitelské stanice je tzv. *agent*. Mimojiné sdružuje všechna data, určuje oprávnění přístupu k nim a případně je poskytuje přes protokol SNMP jiným agentům, nebo aplikacím pro vzdálené ovládání (*management station*). Obvykle sám o sobě zpřístupňuje některé informace o hostitelském systému, jako je například jméno stanice, operační systém a pod. Rozsah dat, které poskytuje, je rozšiřitelný skrze externí aplikace. Právě této vlastnosti je vhodné využít pro zahrnutí vlastních monitorovaných statistik a získat tak systém poskytující data přes protokol SNMP. Další informace o architektuře SNMP jsou zmíněny v kapitole 3.2.

3.1.6 Výsledný návrh

Výsledný návrh respektuje všechny dílčí závěry z předchozích kapitol. Jedná se o systém, jehož funkčnost je komponována z více aplikací. To mu zajišťuje snadnou modifikovatelnost a rozšiřitelnost.

Jak je patrné na obrázku 3.1, základem systému je databáze, která slouží jako úložiště a zdroj všech monitorovaných dat. Šipky na obrázku naznačují směr toku dat. Spodní část obrázku vyjadřuje periorický sběr statistik na hostitelském systému – jednou za časový



Obrázek 3.1: Schéma sytému

interval je aplikací CRONd spuštěn skript pro sběr statistik, který na základě toho, která data jsou monitorována, spustí specializované skripty, z nichž každý zajistí kolekci jedné ze statistik. Přitom tyto skripty využívají prostředky operačního systému, například virtuální adresář `/proc`, systémové utility jako např. `iptables` a pod. Skript pro sběr statistik data následně uloží do databáze, kde jsou připravena být kdykoliv na požádání klienta zpracována.

Horní část obrázku 3.1 vyjadřuje tok dat z databáze ke klientovi. Data mohou být požadována dvěma způsoby:

1. Přes webové rozhraní
2. Prostřednictvím protokolu SNMP

V prvním případě jsou data vybrána z databáze a jsou nejprve zpracována pomocí softwarového modulu `mod_php`, následně jsou odeslána klientovi prostřednictvím webserveru Apache.

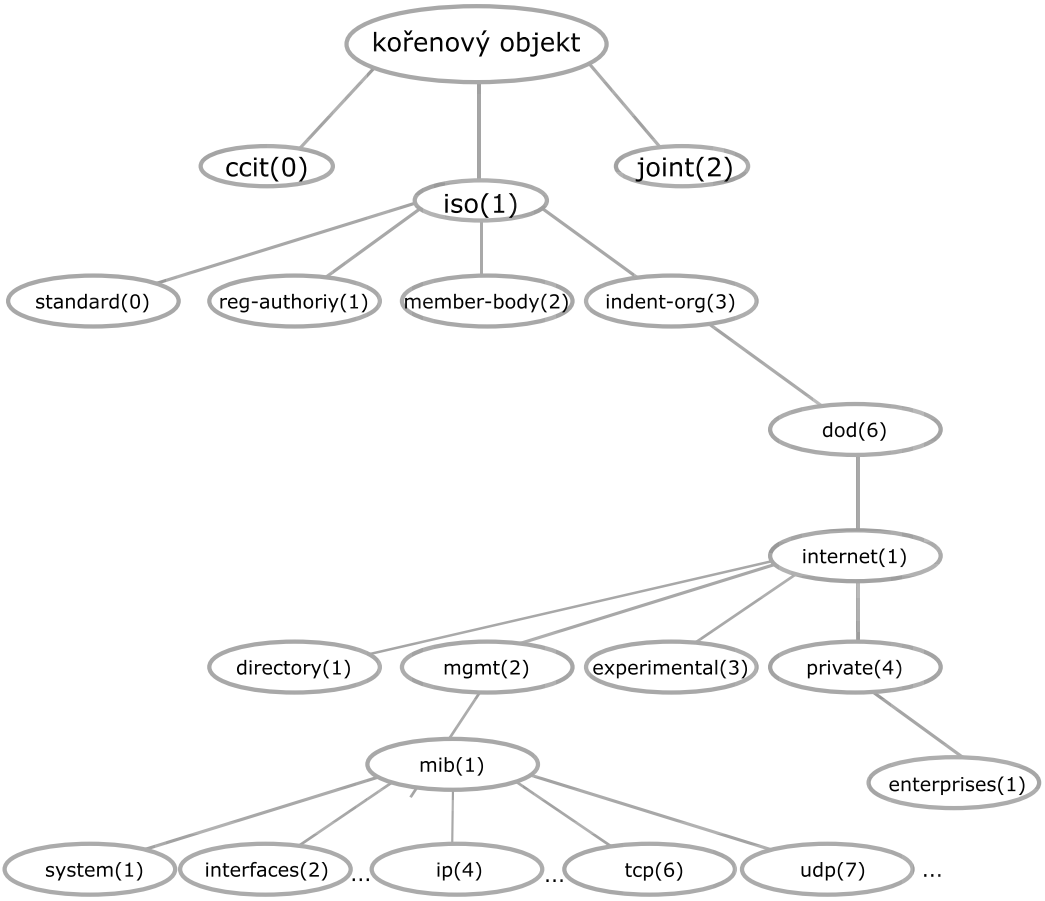
V druhém případě jsou data z databáze vybrána prostřednicvtím skriptu a jsou v požadované formě předána Agentské části balíku `NET-SNMP`, `SNMPd`. Ten je následně poskytne vzdáleným klientům skrze protokol SNMP.

Tato architektura má výhodu v tom, že poskytování dat přes protokol SNMP a webové rozhraní jsou na sobě nezávislé. Proto může být webová část systému funkční i bez balíku `NET-SNMP`, a to i přes to, že poskytované statistiky jsou jednotné.

Vzhledem k tomu, že definice monitorovaných dat za pomocí SNMP je propracovaná, je v systému využita jednotně pro obě rozhraní předávající statistiky klientům. Výběrem statistik se zabývají kapitoly 3.2 a 3.3 na straně 12 resp. 15.

3.2 Výběr SNMP statistik

Jak bylo prezentováno v kapitole 3.1.6 na straně 10, statistiky jsou voleny jednotně jak pro poskytování na webovém rozhraní, tak pro poskytování skrze SNMP.



Obrázek 3.2: základní struktura MIB stromu

Protokol SNMP je dnes hlavním nástrojem pro potřeby správy sítí. Je definován RFC (*Request For Comment*, doporučení pro tvorbu) číslo 1157 [10]. Je navržen tak, aby byl rozšiřitelný skrze tzv. MIB (*Management Information Base*) [13], což je jmenný popis elementů v zařízení a s ním souvisejících dat organizovaný do hierarchické struktury. Každý uvažovaný objekt v zařízení (*managed object*, např.počet odeslaných bajtů na síťovém rozhraní, zatížení procesoru a pod.) má v MIB svoje unikátní číslo, tzv. OID (*Object ID*). Všechny MIB jsou sdruženy do struktury N-árního stromu, jak je patrné na obrázku 3.2.

Firmy využívající SNMP protokol pak mohou pomocí MIB nadefinovat svoje rozšíření, například statistiky spojené s novým typem služby poskytované na síti a pod. Protože je protokol SNMP hojně využíván v malých, ale i rozsáhlých síťových infrastrukturách, je dostupných mnoho veřejných MIB, které je možné využít jako referenci. Při volbě statistik jsem vycházel právě z těchto existujících MIB [4].

Každý uvažovaný spravovatelný objekt v MIB je buď skalární (*scalar object*), nebo tabulkový (*tabular object*). Zatímco skalární objekt je konkrétní instance jednoho objektu, tabulkový objekt je kompozice souvisejících instancí objektů sdružených do MIB tabulky. Například skalárním objektem jsou data odeslaná síťovou kartou eth1 a tabulárním odeslaná data na (obecném) síťovém rozhraní.

MIB používá pro definici datových typů podmnožinu z normy ASN.1 (*Abstract Syntax Notation version 1*). Tato norma byla navržena tak, aby do maximální možné míry neutralizovala nekomaptibility mezi různými monitorovanými zařízeními.

V SNMP verzi 1 jsou využívány následující datové typy (*application-wide data types*):

- `network address` reprezentuje síťovou adresu, SNMP verze 1 podporuje pouze 32 bitů dlouhou adresu.
- `counter` představuje celočíselné počítadlo, je využíván u nezmenšujících se informací, například množství přenesených dat
- `gauge` je celočíselný ukazatel, který má vždy maximální hodnotu z daného intervalu, je vhodný pro indikaci špiček v provozu; v monitorovaném systému udává špičku dané statistiky v období mezi dvěma intervaly měření
- `time tick` představuje setiny sekundy od určité události
- `opaque` reprezentuje řetězec znaků
- `integer` je celé číslo se znaménkem
- `unsigned integer` je celé číslo bez znaménka

Jak plyne z kapitoly 2.2 a ze zadání, výběr statistik by měl zohlednit především přístup klientů ke službě. Datový typ monitorované entity musí odrážet typ poskytovaných dat, při použití existující MIB se definice řídí jí. Umístění v MIB stromu závisí na konkrétní použité MIB.

3.2.1 Výběr statistik pro službu WWW

O přístupu ke službě WWW vypovídá především počet obsloužených klientů, a to jak pro protokol HTTP, tak HTTPS (*HTTP Over SSL*, zabezpečená verze protokolu HTTP). O charakteru obsluhy z hlediska datového objemu jednotlivých požadavků vypovídá množství přenesených dat webserverem. Z hlediska doby setrvání klientů na stránkách se jako vhodná jeví statistika počtu právě aktivních session a počtu právě aktivních TCP spojení. Vzhledem k tomu, že řada stránek je generována dynamicky, je vhodné zařadit i statistiku o tom, kolik času procesoru využívá aplikace webserveru.

Tabulka 3.1 znázorňuje vybrané MIB objekty pro službu WWW.

jméno	datový typ	modul
`wfHttpSummaryOutBytes`	`counter`	WF-HTTP-MIB
`wfHttpSummaryResponses`	`counter`	WF-HTTP-MIB
`cceHttpPerfCpuLoad`	`integer`	CISCO-CONTENT-ENGINE-MIB
`alHttpStatsActiveSessions`	`gauge`	ALTIGA-HTTP-STATS-MIB
`alHttpStatsActiveConnections`	`gauge`	ALTIGA-HTTP-STATS-MIB

Tabulka 3.1: přehled vybraných existujících MIB objektů pro službu WWW

3.2.2 Výběr statistik pro službu FTP

U služby poskytující přenos souborů o stavu vypovídá velmi dobře počet přenosů v obou směrech. Datový charakter souborů, podobně jako u služby WWW, lze poměrně dobře popsat množstvím přenesených dat. Z použití počtu úspěšných přihlášení si lze udělat obrázek o tom, kolik souborů je v průměru přenášeno na jednoho klienta.

Tabulka 3.2 shrnuje vybrané statistiky pro službu FTP.

jméno	datový typ	modul
`wfFtpInFiles`	`counter`	Wellfleet-FTP-MIB
`wfFtpOutFiles`	`counter`	Wellfleet-FTP-MIB
`wfFtpLogins`	`counter`	Wellfleet-FTP-MIB
`alFtpClientStatsOctetsXmit`	`counter`	ALTIGA-FTP-STATS-MIB
`alFtpClientStatsOctetsRecv`	`counter`	ALTIGA-FTP-STATS-MIB

Tabulka 3.2: přehled vybraných existujících MIB objektů pro službu FTP

3.2.3 Výběr statistik pro síťové rozhraní eth0

Síťové rozhraní eth0 poskytuje testovanému serveru konektivitu k internetu. Přes něj jsou tedy směrována data po překladu adres (NAT) od klientů na lokální síti. O překladu adres svědčí především množství přenesených dat a počet směrovaných paketů v síti. Vzhledem k tomu, že jsou přes toto rozhraní poskytovány i služby do internetu, je vhodné monitorovat i celkové datové přenosy na tomto rozhraní. Počet přijatých chybných paketů a zahozených paketů může být zajímavý z hlediska kvality konektivity.

Tabulka 3.3 shrnuje vybrané statistiky pro síťové rozhraní eth0.

jméno	datový typ	modul
`ifOutOctets`	`counter`	IF-MIB
`ifInOctets`	`counter`	IF-MIB
`ifInDiscards`	`counter`	IF-MIB
`ifInErrors`	`counter`	IF-MIB
`ipForwDatagrams`	`counter`	IP-MIB
`nwIpFwdIfCtrFwdBytes`	`counter`	CTRON-IP-ROUTER-MIB

Tabulka 3.3: přehled vybraných existujících MIB objektů pro síťové rozhraní eth0

3.2.4 Výběr statistik pro síťové rozhraní eth1

Na síťovém rozhraní eth1 jsou poskytovány služby klientům lokální sítě. U nich byla v testovaném systému použita kvalitní síťová konektivita, a proto není třeba monitorovat chyby a zahozené pakety. Stejně tak není na toto rozhraní prováděn překlad adres, a proto jsou sbírány pouze statistiky o datových přenosech, jak uvádí tabulka 3.4.

jméno	datový typ	modul
`ifOutOctets`	`counter`	IF-MIB
`ifInOctets`	`counter`	IF-MIB

Tabulka 3.4: přehled vybraných existujících MIB objektů pro síťové rozhraní eth1

3.3 Definice nové MIB pro protokol CIFS

Předchozí kapitola se zabývala výběrem statistik z existujících MIB. Pro službu fileserveru přes protokol CIFS ale není dostupná žádná MIB, a proto je v tomto případě vhodné využít modulárních možností protokolu SNMP a rozšířit definici statistik o tento protokol. Ve stromu MIB budou tyto statistiky umístěny v experimentální větvi .1.3.6.1.3 (viz větev `iso.indent-org.dod.internet.exprimental` v obrázku 3.2 na straně 12), aby bylo patrné, že se nejedná o všeobecně uznávanou definici.

Jak bylo řečeno v kapitole 3.2 na straně 12, rozšíření MIB lze vytvořit za pomocí popisu s použitím normy `ASN.1`. Tento popis má za úkol definovat entity monitorovaného systému, jejich jmenné i číselné idendifikátory a datové typy.

Pro službu fileserveru je z hlediska pohledu na přístup ke službe směrodatná statistika ukazující počet přístupů klientů. Charakteristiku požadavků lze vyjádřit za pomocí statistiky množství přenesených dat. Vzhledem k tomu, že od klientů lokální sítě, kde je služba poskytována, lze očekávat datově náročné přenosy, mohla by být užitečná i statistika zatížení procesoru procesem realizujícím fileserver.

Následující tabulka 3.5 shrnuje vybrané statistiky, které byly sdruženy do nového MIB modulu pod názvem `CIFS-MIB`.

jméno	datový typ	modul
`CIFSSummaryOutBytes`	`counter`	CIFS-MIB
`CIFSSummaryOutPkts`	`counter`	CIFS-MIB
`CIFSSummaryInBytes`	`counter`	CIFS-MIB
`CIFSSummaryInPkts`	`counter`	CIFS-MIB
`CIFSInFiles`	`counter`	CIFS-MIB
`CIFSOutFiles`	`counter`	CIFS-MIB
`CIFSPerfCPULoad`	`integer`	CIFS-MIB

Tabulka 3.5: definice nových MIB objektů pro CIFS

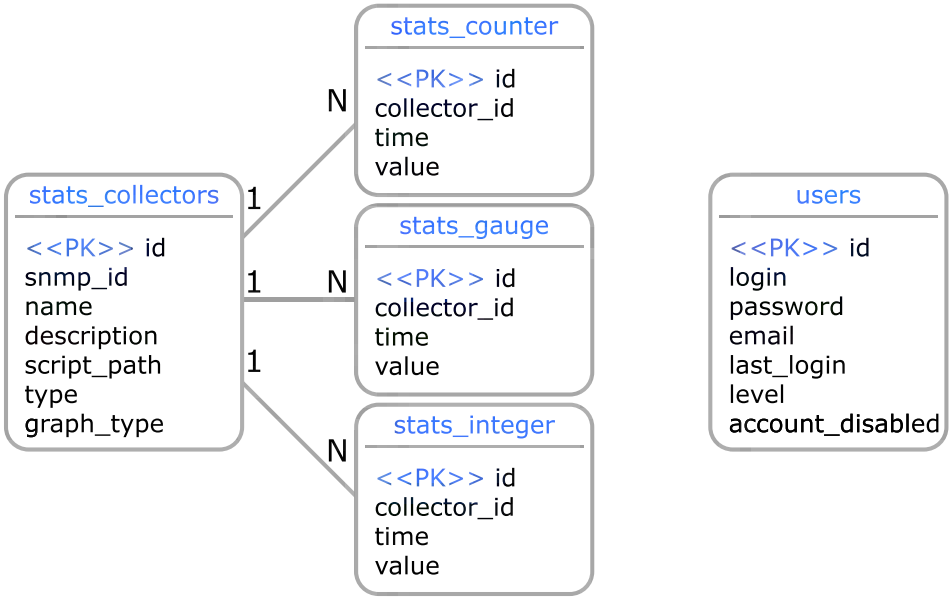
Kompletní definice nové MIB je připojena v příloze A.

3.4 Návrh databáze

Schéma databáze musí umožňovat realizaci všech dílčích požadavků, které jsou uvedeny výše. Jedná se o:

- ukládání informací o statistikách
- ukládání hodnot jednotlivých statistik
- ukládání informací o uživatelích systému

Mimo výše uvedené je třeba ukládat také logovací informace, které se dají využít k ladění funkcionality a celkovému přehledu a o stavu sytému. Tyto informace by ale měly být dosažitelné i při špatném nastavení přístupu k databázi, nebo jiné chybě vedoucí k nedostupnosti databáze. Nad logovacími informacemi není také potřeba vykonávat statistické vyhodnocení a jiné časově náročné operace. Z těchto důvodů je vhodnější směrovat logovací informace spíše do souboru, než do databáze.



Obrázek 3.3: ER diagram databáze systému

Na obrázku 3.3 je uveden ER diagram (*Entity-Relationship*, diagram znázorňující entity a vztahy mezi nimi) schématu databáze implementovaného systému. V pravé části se nachází tabulka `users` s informacemi o uživatelích systému, která nemá žádný vztah s ostatními tabulkami diagramu. V levé části je patrná tabulka `stats_collectors`, která obsahuje informace příslušící každé statistice, jako například jméno, cestu ke skriptu realizující kolekci této statistiky, typ grafu a pod. Na tuto tabulku vztahem 1:N navazují tabulky `stats_counter`,`stats_gauge` a `stats_integer` se samotnými hodnotami statistik. Důvodem rozdělení hodnot statistik do třech tabulek je jejich různý datový typ, který by v případě realizace jedinou tabulkou vedl k nehospodárnému ukládání dat do databáze.

Kapitola 4

Implementace a testování systému

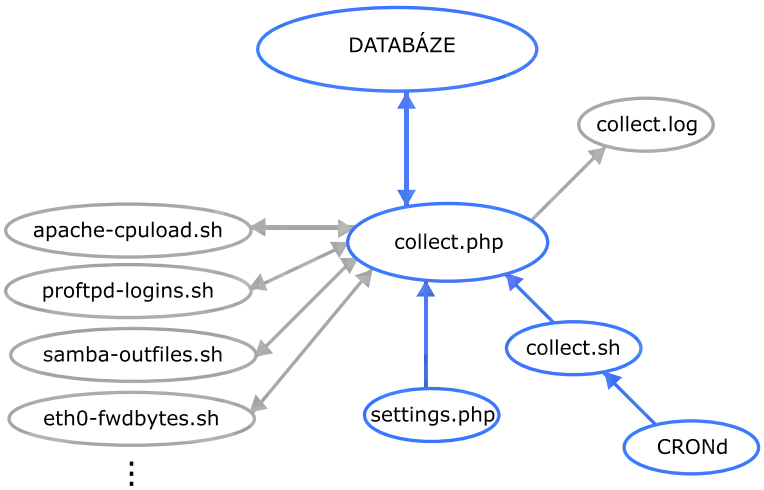
Tato kapitola čtenáři vysvětluje způsob implementace návrhu systému i problémy, které při ní vznikly.

4.1 Jádro systému pro sběr statistik

Implementace jádra systému pro sběr statistik byla realizována na základě návrhu (kapitola 3 na straně 7). Systém sběru dat využívá skriptovacích jazyků, periodického spouštění pomocí aplikace CRONd a databáze. Hlášení o své činnosti směřuje do logovacího souboru.

Sběr statistik je realizován dvěma fázemi:

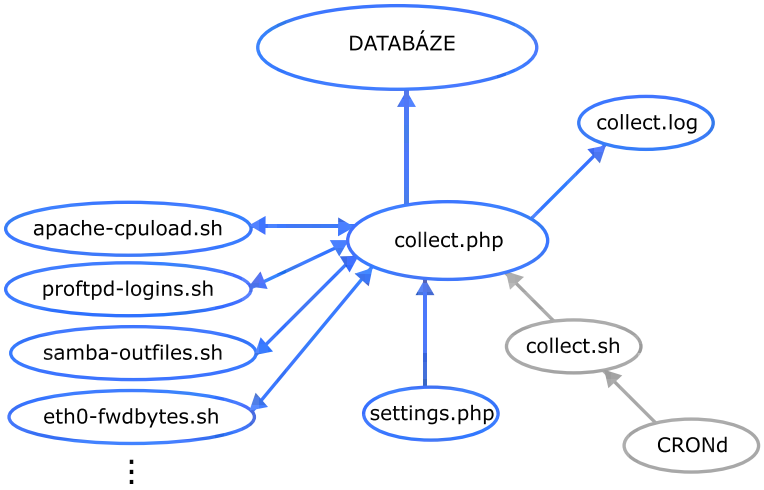
1. Výběr informací o statistikách z databáze
2. Samotná kolekce statistik



Obrázek 4.1: První fáze sběru statistik

Jak je na obrázku 4.1 patrné, v první fázi je spuštěn aplikací CRONd skript `collect.sh`. Ten má za úkol spustit interpretr jazyka PHP a odfiltrovat jeho případné chybové hlášení. Interpretrem jazyka PHP je následně zpracován a vykonán skript `collect.php`, který pro svou činnost využívá soubor `settings.php`. Soubor `settings.php` obsahuje data potřebná

k připojení k databázi a další konfiguraci systému. Za pomocí nich se skript `collect.php` připojí k databázi a vyžádá si od ní informace o statistikách, které má sbírat. Jedná se především o umístění skriptů, které spouští.



Obrázek 4.2: Druhá fáze sběru statistik

Obrázek 4.2 demostruje druhou fázi sběru statistik. Po první fázi, kdy skript `collect.php` získal informace o statistikách, které má sbírat, tyto informace využije k sériovému spouštění konkrétních skritpů pro sběr statistik, například `apache-cpuload.sh`, `proftpd-logins.sh` a pod. Každý z těchto skriptů má svou vlastní konfiguraci potřebnou pro kolekci jeho vlastních dat a vrací vždy hodnotu dané statistiky v období od posledního spuštění do aktuálního spuštění. Tyto informace jsou sbírány skriptem `collect.php`, který je následně ukládá do databáze. O výsledcích operace podává hlášení do logovacího souboru `collect.log`.

Jak již bylo řečeno na úvodu kapitoly, systém sběru statistik využívá pro periodické spouštění aplikace CRONd. Tato aplikace se řídí konfigurací, která je upravitelná administrátorem systému pomocí příkazu `crontab -e`. Sběr statisik také vyžaduje, aby byly správné relativní cesty k soborům sktipů a logovacích souborů vůči skriptu sbírajícímu statistiky. Proto je nutné při spouštění skriptu využít příkazu pro změnu aktuálního adresáře, `cd`.

`*/`N `* * * * cd <`*umístění systému*`> && collect.sh`

Tento příkaz v konfiguračním souboru aplikace CRONd zajistí, že statistiky budou automaticky sbírány na základě konfigurace v databázi každých N minut.

Jak již nastínila kapitola 3.1.3 na straně 8, časový interval pro sběr musí být volen vhodně, aby příliš častá kolekce dat zbytečně nezatěžovala monitorovaný systém a přitom aby data byly dostatečně aktuální. V průběru implementace ale vyšlo najevo, že systémová utilita `ifconfig` využívaná pro monitorování přenosu dat má jako počítadlo pouze 32bitů dlouhé číslo, což při velkém síťovém provozu může vést za časový interval N minut k dvojnásobnému přetečení a tudíž k chybě. Toto je tedy další faktor, který volba časového intervalu mezi sběrem statistik musí zohlednit.

Maximální zachytitelnou rychlost pro 32 bitů dlouhou proměnnou vypočítáme ze vztahu:

$$V_{max} = \frac{2^{32} - 1}{60 \cdot x}$$

kde x je počet minut mezi sběrem statistik.

interval mezi sběry dat	maximální průměrná rychlost
1 minuta	71,6 MB/s
2 minuty	35,8 MB/s
3 minuty	23,9 MB/s
4 minuty	17,9 MB/s
5 minut	14,3 MB/s
6 minut	11,9 MB/s
7 minut	10,2 MB/s
8 minut	8,9 MB/s
9 minut	8,0 MB/s
10 minut	7,1 MB/s

Tabulka 4.1: Vztah intervalu sběru dat vůči maximální zachytitelné průměrné přenosové rychlosti u 32bitů dlouhé proměnné

Vzhledem k tomu, že lokální síť je připojena rychlostí 100Mbit/s, maximální dosažitelná rychlost přenosu u paketů dlouhých průměrně 1500B[14] (tj. standardní *Maximum Transmission Unit*, MTU pro 100Mbit síť) v MB/s je:

$$
\begin{aligned}
V_{LAN} &= \frac{100 \cdot 10^6}{8 \cdot 10^6} \cdot U_{LAN} \\
V_{LAN} &= \frac{100 \cdot 10^6}{8 \cdot 10^6} \cdot \frac{K \cdot B}{K \cdot B + (e-1) \cdot 512} \\
V_{LAN} &= \frac{100 \cdot 10^6}{8 \cdot 10^6} \cdot \frac{1 \cdot 12000}{1 \cdot 12000 + (e-1) \cdot 512} \\
V_{LAN} &\doteq 12,5 \cdot 0,93 \\
V_{LAN} &\doteq 11,625
\end{aligned}
$$

Kde

- U_{LAN} je utilizace lokální sítě.
- B je průměrná délka paketů v bitech
- K je počet paketů za cyklus

Z tabulky 4.1 plyne, že nejdelším intervalem pro maximální hodnotu 11,625MB/s je 6 minut. Vzhledem k tomu, že data v systému musí být dostatečně aktuální, jeví se vhodnější kompromis sběru dat jednou za 5 minut.

4.2 Skripty pro sběr statistik

4.2.1 Společné vlastnosti a problémy

kapitola 3.2 na straně 12 zmiňovala využívání datových typů v jednotlivých statistikách. Zatímco datové typy jako `integer` nebo `gauge` definující statistiku jako poslední hodnotu,

resp. nejvyšší hodnotu mezi intervaly, datový typ `counter` definuje danou statistiku jako neklesající, čímž vyvolává řadu na sebe navazujících problémů.

Situaci si lze demonstrovat na ukazateli množství přenesených dat na síťovém rozhraní. Tato hodnota se bude s časem stále zvětšovat, čímž dojde k několika problémům:

- Jednak ukazatel v systému, který tuto statistiku reprezentuje bude dříve či později vynulován, ať už restartem systému, nebo přetečením proměnné, která jej reprezentuje. V případě, že by celá hodnota čítače byla periodicky ukládána by toto vedlo k chybnému vracení výsledků od doby nulování.
- Dalším problémem je periodické ukládání této hodnoty do databáze. Vzhledem k tomu, že každá následující hodnota bude v sobě obsahovat i velikost hodnoty předchozí, enormě se tím zvyšuje náročnost na rozsah datového typu při ukládání a s ním se zvětšuje i množství dat, které bude každý časový interval sběru ukládáno. Navíc by velikost datového typu ovlivňovala dobu, po kterou může být statistika sbírána, což je krajně nevhodné.
- Problematické je v tomto případě i statistické zpracování pro přehled o dlouhodobějším stavu statistiky, protože není možné jednoduše používat dostupné databázové funkce jako `MAX()` nebo `SUM()`.

Z těchto důvodů je vhodnější ukládat místo aktuální absolutní hodnoty čítače pouze relativní přírůstek vůči minulé hodnotě dané statistiky. Problém s přetečením čítače se dá řešit uložením poslední absolutní hodnoty dané statistiky. Jestliže je nová absolutní hodnota menší, než předchozí, lze za předpokladu, že čítač přetekl pouze jednou, vypočítat správnou hodnotu ze vztahu:

$$X_{new} = C_{maxval} - X_{old} + X_{inc}$$

Kde

- X_{new} je nová, správná hodnota přírůstku čítače
- C_{maxval} je maximální hodnota, kterou čítač proměnné X může nabývat
- X_{old} je minulá absolutní hodnota čítače
- X_{inc} je přírustek čítače mezi posledním a aktuálním měřením

Tímto způsobem lze účinně obcházet přetečení systémových čítačů a sbírat tak správně statistiky.

Od předpokladu, že čítač dané proměnné přetekl pouze jednou se odvíjí i údaje o maximální průměrné přenosové rychlosti z tabulky 4.1.

4.2.2 Sběr statistik služby WWW

Služba www je poskytována, jak již bylo v kapitole 2.1 řečeno, skrze aplikaci `Apache`. Vzhledem k velké rozšířenosti tohoto webserveru je pro něj dostupná řada modulů, z nichž většina je součástí standardní instalace. Mezi takového moduly patří i `mod_stat`, což je specializovaný nástroj určený přímo k monitorování statistik spojených s tímto webserverem.

Pro svou správnou funkci vyžaduje načtení realizované příkazem v konfiguračním souboru `httpd.conf`:

```
LoadModule mod_stats
```

Mimo tento příkaz je ještě nutné zajistit oprávnění pro přístup ke statistikám tohoto modulu, například tímto způsobem:

```
<Location /server-status>
  SetHandler server-status
  Order Deny,Allow
  Deny from all
  Allow from localhost
</Location>
```

Tyto příkazy zajistí zobrazení statistik pro klienty připojující se z následující adresy:

`http://localhost/server-status?auto`

Relevantní statistiky jsou pak získány separací za pomocí standardních systémových utilit jako je `grep`, `sed`, `cut` a pod. Jedná se o:

- Počet právě aktivních požadavků na obsluhu klienta
  - realizováno skriptem `apache-current_accesses.sh`
- Celkový počet odeslaných požadavků
  - realizováno skriptem `apache-total_accesses.sh`
- Vytížení procesoru serveru procesem webserveru
  - realizováno skriptem `apache-cpuload.sh`
- Celkové množství odeslaných bajtů dat webserverem (pro HTTP i HTTPS dohromady)
  - realizováno skriptem `apache-traffic.sh`
- Počet právě aktivních session
  - realizováno skriptem `apache-current_sessions.sh`

4.2.3 Sběr statistik služby FTP

Pro službu FTP jsou sbírány následující statistiky:

- počet přihlášení klientů - realizováno skriptem `proftpd-logins.sh`
- počet odeslaných a přijatých souborů - realizováno skripy `proftpd-rxfiles.sh` a `proftpd-txfiles.sh`
- množství přenesených dat v obou směrech - realizováno skripy `proftpd-rxbytes.sh` a `proftpd-txbytes.sh`

Aplikace `ProFTPd`, na které je služba realizována, má rozsáhlé možnosti konfigurace a logování, což je výhodné pro sběr dat. Logovací informace jsou zaznamenávány do dvou souborů: systémového logu (*system logfile*) a rozšířeného logu (*extended logfile*). Zatímco systémový log je využíván pro ukládání informací o chybách, startu a zastavení služby, přihlášení uživatelů a pod., do rozšířeného jsou směrovány detailní údaje o přenesených datech a zpracovávaných příkazech od klientů. Oba tyto soubory lze využít pro extrakci potřebných statistik.

K získání informací z logovacích souborů je využito skriptovacího jazyku BASH a standardních systémových utilit operačních systémů typu UNIX, `grep` , `cut`, `bc` a pod.

4.2.4 Sběr statistik služby fileserver

Na základě návrhu v kapitole 3.3 na straně 15 jsou sbírány tyto statistiky:

- počet přijatých a odeslaných souborů - realizováno skripty `samba-infiles.sh` a `samba-outfiles.sh`
- množství přenesených dat z a do serveru - realizováno skripty `samba-inbytes.sh` a `samba-outbytes.sh`
- počet odeslaných a přijatých paketů - realizováno skripty `samba-inpkts.sh` a `samba-outpkts.sh`
- vytížení procesoru monitorovaného serveru aplikací fileserveru - realizováno skriptem `samba-cpuload.sh`

Aplikace fileserveru, `samba`, má, podobně jako `ProFTPd`, dobré možnosti konfigurace a logování. Lze nastavit tzv. loglevel, což je stupeň reprezentující množství informací zasílaných do logu. Číslo stupně a množství logovaných informací jsou v přímé úměře. Nastavení logování se provádí v konfiguračním souboru následující direktivou:

`log level =` N

kde N je výše zmíněné číslo udávající množství informací. Z logovacího souboru je vhodně extrahovat informace o statistikách. Pro $N = 1$ jsou do logu směrovány pouze základní informace o stavu aplikace. Vyšší loglevel, $N = 2$, zajistí monitorování otevírání souborů, ne však množství přenesených dat. Stupeň loglevelu $N = 3$ směruje do logu informace jednotlivých datových přenosů, avšak po jednotlivých vyprázdněních bufferu, což vede k velmi rychlému zvětšování logovacího souboru a zdlouhavé extrakci dat. Proto je vhodné ponechat $N = 2$, soubor logu využít pouze k monitorování přístupu k souborům, a statistiky o přenesených datech a odeslaných paketech získávat jinak.

Systémová utilita `iptables` nabízí pro účely monitorování dat označování paketů za pomocí přepínače `-j`. S přihlédnutím k tomu, že pro služby sdílení souborů a tiskáren jsou standardně využívány pevně definované porty, lze této utility využít pro monitorování přenesených dat a paketů služby fileserver. Pro odchozí data se jedná o příkazy:

`iptables -t mangle -A OUTPUT -p tcp --sport 139 -j MARK --set-mark` N
`iptables -t mangle -A OUTPUT -p tcp --sport 445 -j MARK --set-mark` N

Tedy odchozí data z portů 139 a 445. Pro opačný směr pak:

```
iptables -t mangle -A INPUT -p tcp --dport 139 -j MARK --set-mark N
iptables -t mangle -A INPUT -p tcp --dport 445 -j MARK --set-mark N
```

Tedy příchozí data na porty 139 a 445. Parametr N v tomto případě udává číslo, kterým budou pakety označovány a které je pak následně využito jako identifikace při výběru dat příkazem:

```
iptables -t mangle -L -v -x -n
```

Za pomocí utilit jako `grep`, `cut`, `bc` a pod. lze konkrétní data extrahovat z tohoto výpisu, čehož je využito ve výše uvedených skriptech.

4.2.5 Sběr statistik síťových rozhraní

Pro síťová rozhraní jsou monitorovány následující statistiky:

- množství přenesených dat v obou směrech - realizováno skripty `eth0-tx.sh` a `eth0-rx.sh` pro síťové rozhraní `eth0`, resp. `eth1-tx.sh` a `eth1-rx.sh` pro síťové rozhraní `eth1`
- množství zahozených paketů - realizováno skriptem `eth0-dropped.sh`
- množství chybných paketů - realizováno skriptem `eth0-errors.sh`
- možství paketů, které prošly službou NAT - realizováno skriptem `eth0-fwdpackets.sh`
- možství dat, která prošla službou NAT - realizováno skriptem `eth0-fwdbytes.sh`

Pro správu síťových rozhraní je v operačních systémech typu UNIX hojně využívána systémová utilita `ifconfig`. Ta, kromě konfigurace síťových rozhraní, umožňuje i zobrazovat statistiky o jejich provozu. Vzhledem k tomu, že je běžně dostupná, je vhodné ji pro monitorování využít. Čítač v ní obsažený má bohužel pouze 32bitů, a proto trpí problémem častého přetečení, jak je uvedeno v kapitole 4.2.1 na straně 19. Při použití ukládání poslední absolutní hodnoty čítače a dostačně krátkého intervalu sběru dat je utilita `ifconfig` plně dostačující i pro rychlosti lokální sítě. Datové přenosy a počty paketů v rámci jednoho rozhraní jsou tedy monitorovány skrze tuto utilitu.

Monitorování služby NAT není zahrnuto v systémové utilitě `ifconfig`, a proto je třeba najít jinou alteranativu. Pro nastavení firewallu, směrování a označování dat je využívána utilita `iptables`, která v sobě také obsahuje možnosti monitorování. Pro účely NAT má k dispozici speciální tabulku, která zahrnuje tyto přenosy. Následující příkaz lze použít k zobrazení směrovaných dat a paketů:

```
iptables -t nat -L -v -x
```

S použitím systémových utilit zmiňovaných v předchozích kapitolách lze informace jak z utility `ifconfig`, tak z utility `iptables` extrahovat a využít pro monitorování.

4.3 Poskytování statistik přes SNMP

Jak již bylo zmíněno v kapitole 3.1.5 na straně 10, systém poskytuje statistiky přes protokol SNMP s využitím balíku aplikací `NET-SNMP`, konkrétně pak aplikace `SNMPd`. Tato aplikace má možnost rozšířit poskytované statistiky skrze direktivu v konfiguračním souboru :

`pass` *MIBOID příkaz parametry*

Kde MIBOID je unikátní identifikátor monitorované entity v MIB stromu a příkaz s parametry představuje cestu ke skriptu, který je spuštěn s danými parametry, kdykoliv je aplikací `SNMPd` přijat dotaz na danou entitu. Tento příkaz musí vrátit data v přesně definované formě, jinak nebudou akceptována:

MIBOID
datový typ
data

Kde

- *MIBOID* je identifikátor dané entity v MIB stromu (stejný, jaký je uveden v konfiguračním souboru `SNMPd` pro direktivu `pass`)
- *datový typ* je jeden z identifikátorů blíže popsaných v kapitole 3.2 na straně 12
- *data* jsou hodnota vrácená skriptem, například počet přenesených bajtů dat v časovém období a pod.
- jednotlivé informace jsou odděleny znakem nového řádku

Pro automatické generování konfigurace je v implementovaném systému určen skript `gensnmpdconf.php`, který je možné aktivovat skrze webové rozhraní systému. Tento skript se po spuštění na základě informací v souboru `settings.php` připojí k databázi, získá informace o monitorovaných statistikách a s respektováním syntaxe konfiguračního souboru aplikace `SNMPd` vygeneruje konfigurační soubor. Název generovaného souboru se odvíjí od konfigurace v souboru `settings.php`, konkrétně proměnné `$SNMP_CONFIG_FULLPATH`. Standardně je tento název souboru `snmp.conf.inc` v adresáři, kde se nachází monitorovací systém.

Po generování rozšiřující konfigurace pro aplikaci `SNMPd` je třeba ji znovu spustit s parametrem `-c` a umístěním nového souboru:

`snmpd -c <`*umístění systému*`>/snmpd.conf.inc`

Tímto příkazem se zajistí spuštění aplikace `SNMPd` s rozšířeními MIB stromu definovanými implementovaným systémem. Konfiguraci oprávnění přístupu, naslouchání na síťových rozhraních a pod. je třeba nastavit v jednom ze standardních konfiguračních souborů aplikace [3]. K tomu lze využít skriptu dodávaného s balíkem `NET-SNMP`, `snmpconf`.

Vzhledem k tomu, že monitorovací systém pouze poskytuje monitorované data skrze protokol `SNMP` (umožňuje realizaci příkazu `read`), jeví se využití protokolu verze 1 jako plně dostačující.

V kapitole 3.3 na straně 15 byla popsána definice nové MIB pro protokol CIFS. Aby byla tato definice používána systémem NET-SNMP, je třeba provést následující příkazy:

```
cp <umístění systému>/CIFS-MIB.txt /usr/local/share/snmp/mibs
echo 'mibs +CIFS-MIB' >> $HOME/.snmp/snmp.conf
```

Funkčnost si lze ověřit například příkazem:

```
snmpget -c readers -v1 <adresa SNMPd> .1.3.6.1.3.255.1.1
```

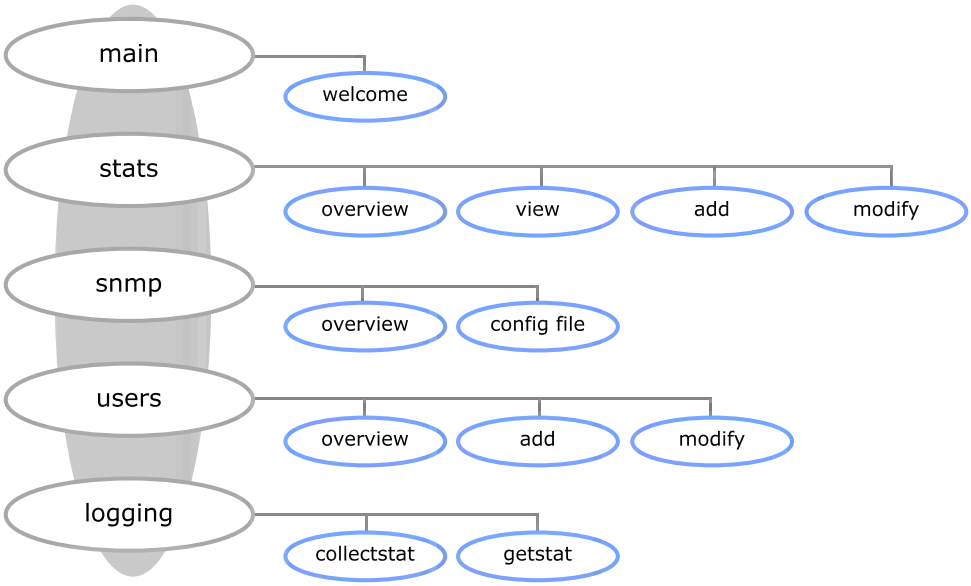
který by měl vrátit jako identifikační řetězec textovou reprezentaci entity:

```
CIFS-MIB::cifsSummaryOutBytes
```

4.4 Webové rozhraní

Základní charakteristika webového rozhraní implementovaného systému byla navržena v kapitole 3.1.4 na straně 9. Část systému dostupná skrze protokol HTML poskytuje především statistiky zpracované z hlediska dlouhodobějšího pohledu, tj. do grafické podoby a s využitím matematických statistických funkcí. Díky možnosti vytvořit uživatelské rozhraní je vhodné i pro administraci celého systému.

4.4.1 Zobrazované informace a systém menu



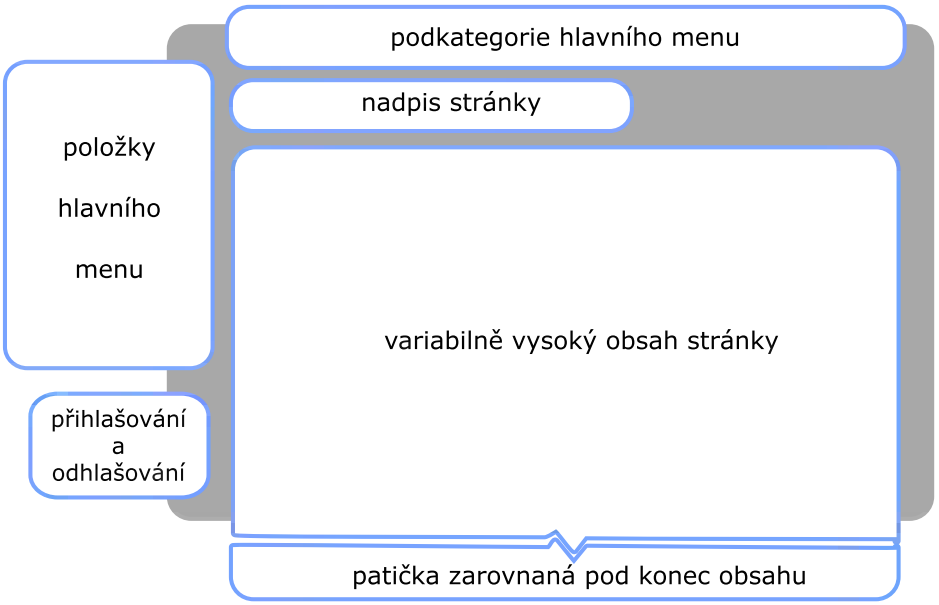
Obrázek 4.3: položky menu na stránce webového rozhraní

S přihlédnutím k uživatelské intuitivnosti byl navržen systém menu, který představuje obrázek 4.3. Jak je na něm patrné, menu je rozděleno do hlavní (šedé položky) a vedlejší kategorie (modré položky), přičemž ve vedlejší kategorii jsou vždy zobrazovány nabídky patřící tématicky do vybrané kategorie hlavního menu. Toto rozdělení postihuje všechny kategorie, které jsou na základě návrhu v systému obsaženy.

4.4.2 Rozmístění prvků na stránce

Rozmístění prvků na stránce webového rozhraní systému by mělo respektovat základní pravidla intuitivnosti a přehlednosti. Vzhledem k tomu, že se v našich zemích čte směrem zleva doprava, je více pravděpodobné, že uživatel se nejdříve podívá na levou stranu rozhraní. V tomto místě by měl narazit na orientační bod, hlavní menu. Vedlejší menu je vhodné realizovat podobně jako záložky, protože ty jsou standardní součástí většiny graficky orientovaných operačních systémů, a proto jsou na ně uživatelé zvyklí. Obě menu by měly být dostečně výrazné, aby přilákaly uživatelovu pozornost. Převážnou část strany by měl zabírat obsah. Dialog pro přihlašování a odhlašování uživatelů by měl být umístěn tak, aby byl vidět na běžném rozlišení obrazovky bez použití posunu stránky. Na místě pod hlavním menu by nebyl příliš do očí bijící, ale přitom na první pohled viditelný. Nadpis stránky by

měl být umístěn mezi záložkovým menu a samotným obsahem stránky. Patička, obsahující záhlaví stránky, by měla být umístěna vždy na konci obsahu nezávisle na jeho délce.



Obrázek 4.4: rozložení prvků na stránce webového rozhraní

Obrázek 4.4 schématicky ukazuje rozmístění prvků webového rozhraní, které vzniklo s respektováním výše uvedených pravidel. Samotná implementace využívá standardů XHML[1] 1.1 a CSS (*Cascading Style Sheet*, stylování vzhledu a rozmístění prvků na stránce pomocí kaskádovacích stylů). Grafický návrh vznikl za pomocí aplikace Adobe Photoshop.

4.4.3 Vyhodnocování statistik z dlouhodobého hlediska

Statistiky jsou webovým rozhraním systému vyhodnocovány z dlouhodobého hlediska dvěma způsoby:

1. Matematickými statistickými funkcemi
2. Grafickou reprezentací hodnot

Z matematických statistických funkcí je vhodné zahrnout především tyto:

- Rozptyl
- Směrodatnou odchylku
- Aritmetický průměr
- Maximum, minimum

[1] eXtensible HyperText Markup Language byl vyvinut jako standard organizací World Wide Web Consortium (W3C) jako nástupce HyperText Markup Language

Rozptyl diskrétní náhodné veličiny je definován vztahem [11]:

$$D(X) = \frac{1}{n}\sum_{i=0}^{n}(x_i - E(X))^2$$

Kde

- $E(X)$ je střední hodnota vzorku dat
- x_i je vzorek dat

Střední hodnotu $E(X)$ v tomto případě lze aproximovat aritmetickým průměrem vyjádřeným vztahem:

$$\overline{x} = \frac{1}{n}\sum_{i=0}^{n}x_i$$

Směrodatná odchylka je pak definována jako odmocnina z rozptylu:

$$\sigma = \sqrt{D(X)}$$

Pro grafické reprezentování hodnot byla využita knihovna `JpGraph`, jak již bylo zmíněno kapitole 3.1.4 na straně 9. Tato knihovna umožňuje jednoduše a rychle vytvářet z dat širokou škálu grafů. V systému je umístěna v adresáři `/include` a je používána skriptem v `graph.php`. Ten je volán funkcí mající na starost zobrazování statistik, parametry jsou předávány prostřednictvím HTTP hlavičky, řádku `GET`. Skript `graph.php` se na základě nastavení v souboru `settings.php` připojí k databázi, získá z ní potřebná data a použije knihovnu na to, aby je vykreslil. Je přitom uzpůsoben tak, že jediná data, která odesílá klientovi webového rozhraní představují případný obrázek, a proto může být ve webovém rozhraní volán HTML direktivou `img`. Tento způsob volání zajišťuje možnost zobrazení uživatelsky srozumitelného hlášení informujícího o tom, že vyhodnocení statistik je teprve generováno a bude chvíli trvat. Doba zpracovávání grafu závisí především na počtu zobrazovaných hodnot, a proto se déle generují grafy sumarizující delší časové období. I pro delší časová období však na testovaném systému nepřesáhla doba generování jednotky sekund, což lze považovat za uživatelsky interaktivní.

4.4.4 Uživatelé systému a oprávnění

S ohledem na základní pravidla bezpečnosti jsou uživatelé systému rozděleni do čtyřech skupin:

- not privileged - neoprávnění uživatelé, kteří nemohou se systémem vůbec manipulovat
- readers - mohou si prohlížet statistiky a stav systému, avšak nemohou nic upravovat ani přidávat
- powerusers - mohou zobrazovat, přidávat a upravovat statistiky i uživatele. Mohou také generovat nový konfigurační soubor `SNMPd`, avšak nemohou si prohlížet systémové logy

- administrators - mohou libovolně manipulovat s celým systémem

Příslušnost k některé z těchto skupin uživatel prokazuje při přihlašování za pomocí uživatelského jména a hesla.

4.5 Testování ve virtuálním prostředí

Implementovaný systém byl v provozu testován na linuxovém serveru poskytujícím síťové služby, jak již bylo zmíněno na začátku této práce v kapitole 2.1 na straně 5. Pro účely testování instalace by ale bylo potřeba nasadit systém na jiný server, a proto je lepší využít virtualizační technologie. Ty jsou velmi vhodné i pro demonstraci funkcionality systému.

Systém byl nainstalován na virtuální stroj Microsoft Virtual PC, a to jak v serverové, tak klientské části. Pro účely demonstrace je pak šířen datovými soubory s příponou .vhd, které reprezentují data uložené na pevném disku počítače. Jelikož jsou tyto soubory poměrně velké (v řádu GB), je pro jejich distribuci využito DVD a komprimační program ZIP.

Virtuální prostředí bylo při implementaci použito pro testování a vytvoření přesného postupu instalace v souboru INSTALL. Díky instalaci na virtuálním prostředí byly také odhaleny chyby způsobené nulovými požadavky na služby, chyby u konfigurace a pod.

Kapitola 5

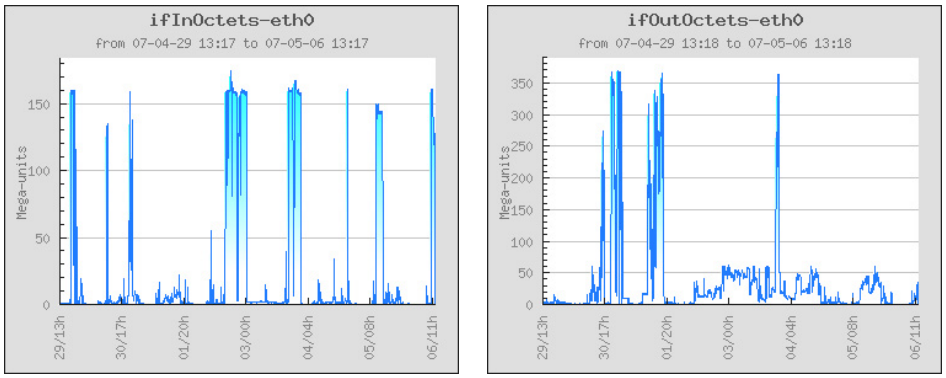
Závěr

Tato kapitola přestavuje závěrečné shrnutí nabytých poznatků během celé práce.

5.1 Analýza výsledků

Implementovaný systém sběru statistik byl nasazen v reálném prostředí po dobu několika týdnů, kdy autonomně sbíral statistiky o serveru. Během této doby byl nasbírán dostatečně velký vzorek dat pro analýzu charakteru statistik.

5.1.1 Statistiky síťových rozhraní



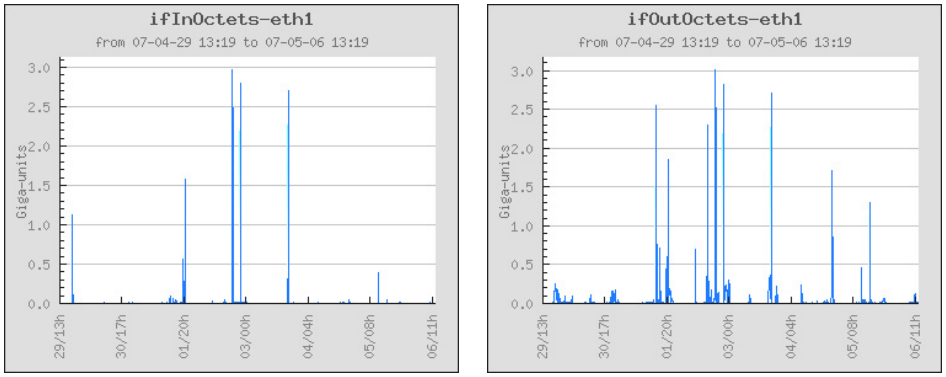
(a) přijatá data

(b) odeslaná data

Obrázek 5.1: přenesená data síťového rozhraní eth0

Grafy přenosu dat (5.1) na síťovém rozhraní eth0, které realizuje konektivitu do sítě internet, vykazují poměrně kolísavé využití linky. U grafu přijatých dat jsou zřejmé špičky, kdy byla linka využívána na svoji plnou kapacitu. U odeslaných dat jsou tyto špičky patrné také, nicméně v grafu jde navíc vidět stále využití požadavky na služby serveru.

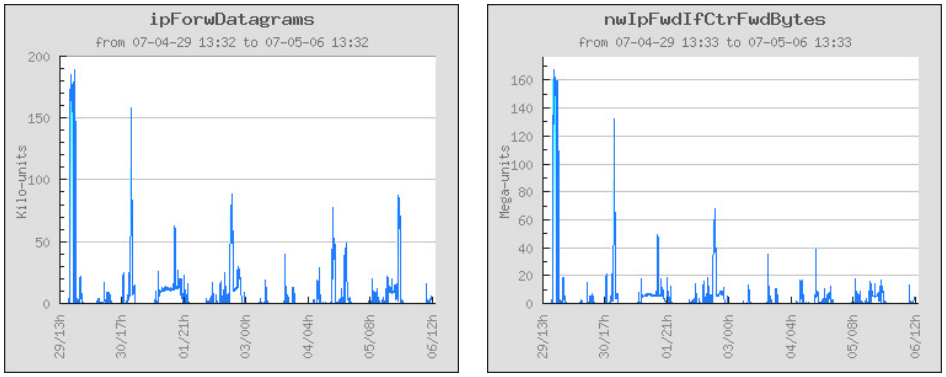
Přenos dat na síťové rozhraní eth1, které je připojeno do lokální sítě, je patrný na grafech 5.2. Na první pohled zaujme velký rozptyl hodnot a dlouhé doby s nízkými hodnotami, způsobené malou aktivitou klientů lokální sítě. V porovnání s grafem 5.1 je zjevné, že



(a) přijatá data (b) odeslaná data

Obrázek 5.2: přenesená data síťového rozhraní eth1

požadavky lokální sítě jsou obsluhovány větší přenosovou rychlostí, avšak pro menší počet klientů.

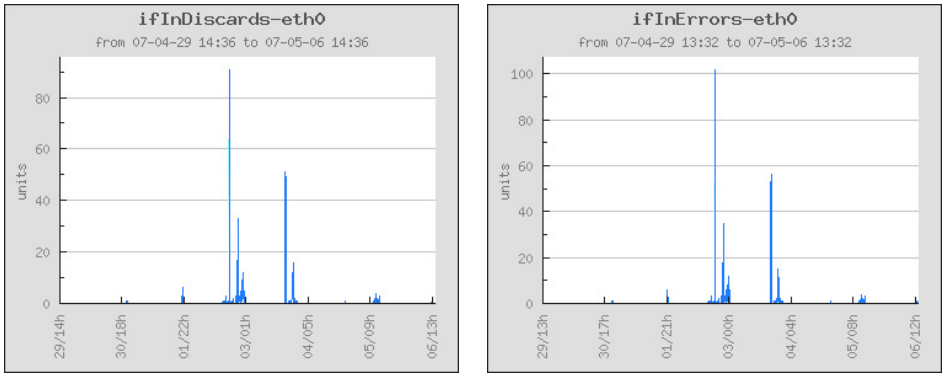


(a) počet směrovaných datagramů (b) množství směrovaných dat

Obrázek 5.3: Překlad adres na síťovém rozhraní eth0

Grafy 5.3 představují přístup klientů lokální sítě do internetu prostřednictvím překladu adres. Při porovnání s grafem 5.1 je patrné, že většina konektitivy přípojky eth0 byla využita pro samotný server, čili pro obsluhu síťových služeb.

Chybovost rozhraní je demonstrována grafy 5.4. Jde vidět, že zahozené pakety a chybné pakety jsou v přímé úměře, avšak jejich počty se sobě nerovnají. Graf celkově vykazuje dvě větší špičky, z čehož lze usuzovat, že v dané době nastaly události způsobující zvýšenou chybovost na vedení.



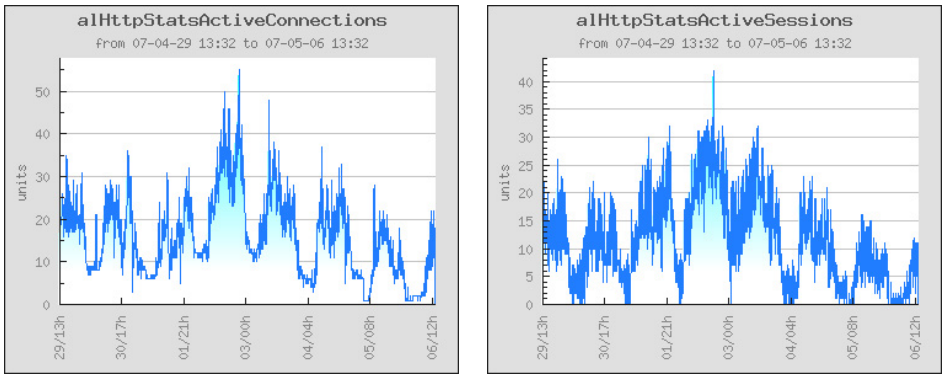
(a) zahozené pakety

(b) chybné pakety

Obrázek 5.4: Chybovost na síťovém rozhraní eth0

5.1.2 Statistiky služby WWW

Na grafech 5.5 reprezentující aktivní připojení klientů na službu WWW je dobře patrný malý rozptyl. Jde vidět, že služba je využívána víceméně neustále a podle očekávání se síťový provoz částečně utlumuje přes večerní hodiny. Přímá úměra mezi hodnotami obou grafů napovídá, že většina přístupů byla realizována s využitím session.

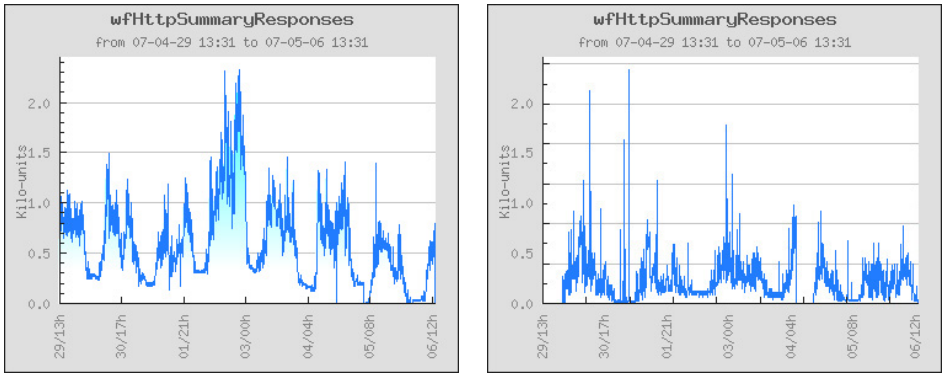


(a) aktivní TCP/IP spojení

(b) aktivní session

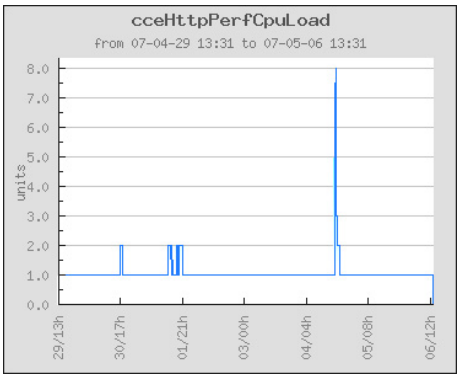
Obrázek 5.5: Aktivní připojení na službu WWW

Aktivita klientů na službě WWW je vyjádřena grafy 5.6. Počet obsloužených požadavků je naprvní pohled podobný aktivním TCP/IP spojení z minulé dvojice obrázků. Z toho vyplývá, že každé síťové spojení vede obvykle na podobný počet HTTP dotazů na webserver. To může být způsobeno chrakterem stránek hostovaných na webserveru. Naproti tomu množství odeslaných dat webserverem má větší rozptyl a celkově více inklinuje k náhlým špičkám. Jednotlivé požadavky tedy nemají příliš podobnou výslednou velikost.



(a) počet obsloužených požadavků (b) množství odeslaných dat

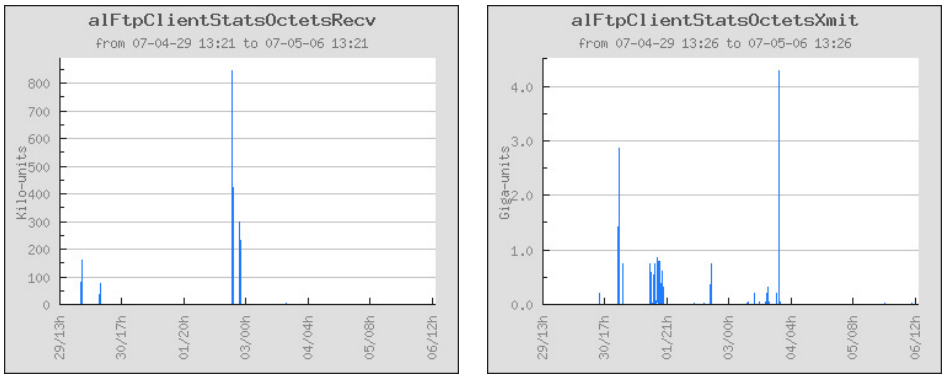
Obrázek 5.6: Aktivita na službě www



Obrázek 5.7: Zatížení procesoru službou webserveru

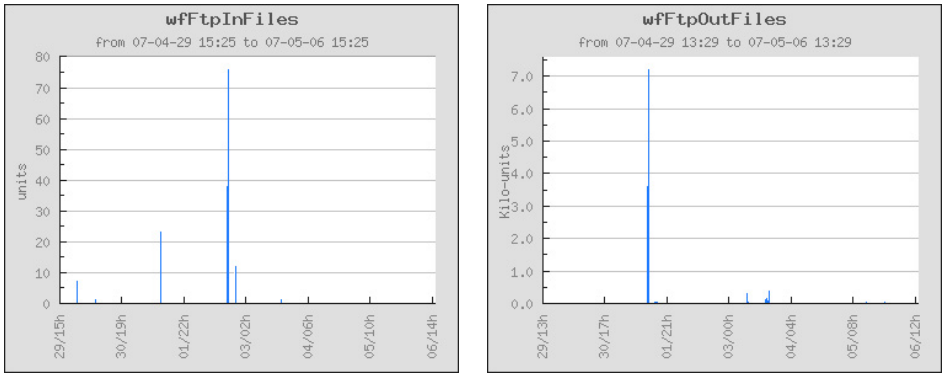
Zatížení procesoru službou www, jak jej ukazuje graf 5.7 jen výjimečně narostlo přes hodnotu 1 %.

5.1.3 Statistiky služby FTP



(a) množství přijatých dat (b) množství odeslaných dat

Obrázek 5.8: Přenos dat na službě FTP

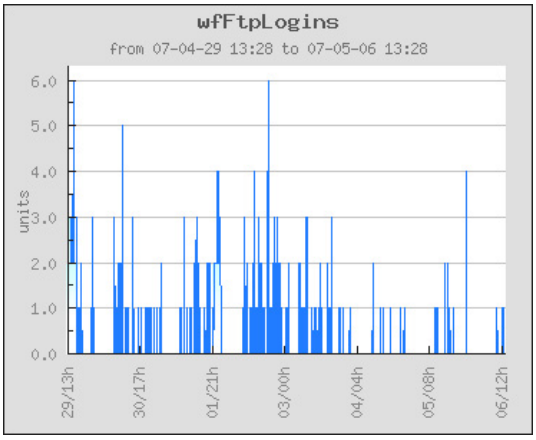


(a) počet přijatých souborů (b) počet odeslaných souborů

Obrázek 5.9: Přenos souborů na službě FTP

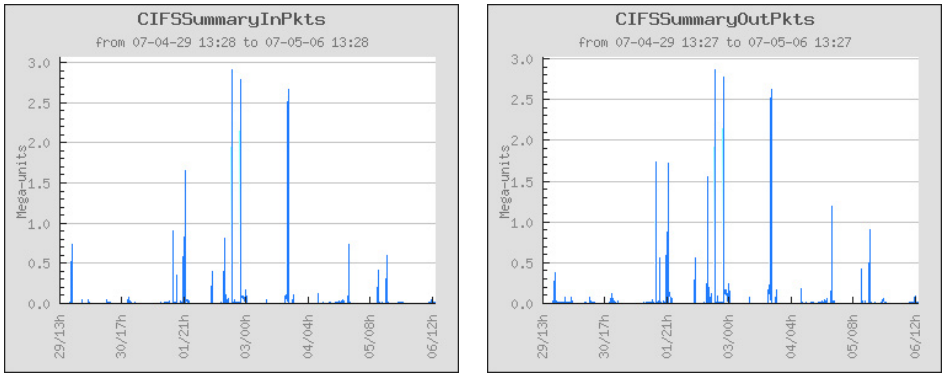
Datové přenosy u služby FTP vyjádřené grafy 5.8 mají zjevný charakter náhlých špiček a velkého rozptylu. Při porovnání s grafy 5.9 vyjadřujícími počet přenesených souborů v obou směrech je ale zřejmé, že operacemi náročnými na přenos dat byl především přenos velkých souborů.

Počet přihlášení klientů, který odráží graf 5.10 je relativně malý vůči množství přenesených dat.



Obrázek 5.10: Počet přihlášení klientů na službu FTP

5.1.4 Statistiky služby fileserver



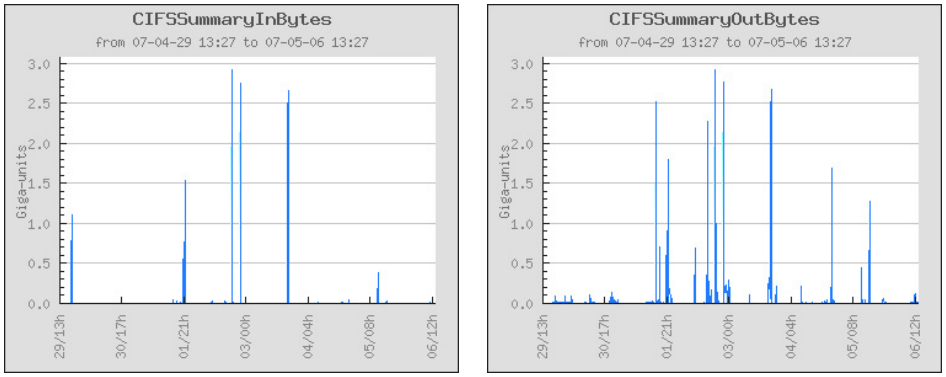
(a) přijaté pakety

(b) odeslané pakety

Obrázek 5.11: Přenos paketů služby fileserveru

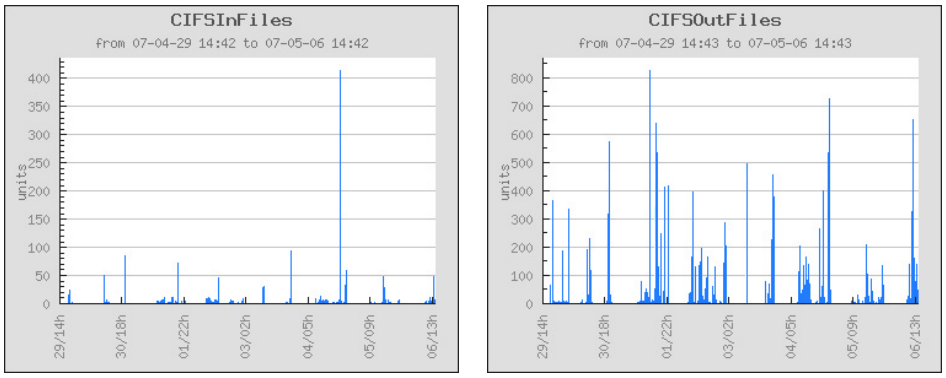
Služba fileserver byla v průběhu testování poskytována výhradně klientům lokální sítě. Tento fakt se také projevil na množství přenesených dat a charakteru odesílání: Jak je patrné z porovnání grafů 5.11 a 5.12, data byla odesílána i přijmána téměř s konstantní délkou paketů, což svědčí o žádném, nebo minimálním směrování a o kvalitní síťové konektivitě v topologii testované lokální sítě. Zároveň je z grafů patrný velký rozptyl hodnot, což je charakteristické pro malý počet klientů přenášejících ojediněle velké objemy dat (například při kopírování souboru).

Z grafů přenosů souborů 5.13 plyne, že fileserver je častěji využíván pro poskytování souborů, než k jejich ukládání.



(a) přijatá data (b) odeslaná data

Obrázek 5.12: Přenos dat služby fileserveru



(a) přijaté soubory (b) odeslané soubory

Obrázek 5.13: Přenos souborů služby fileserver

5.2 Zhodnocení

Během doby, kdy byl systém nasazen v reálném prostředí prokázal svou plnou funkčnost, i praktické využití pro účely monitorování síťového provozu. Nalezne uplatnění všude tam, kde jsou operační systémy typu UNIX využívány k poskytování síťových služeb, především v síťových serverech na internetu. Narozdíl od specializovaných monitorovacích systémů je díky snadné modifikovatelnosti a možnosti přidávání nových statistik snadno adaptovatelný na měnící se prostředí a služby. Rozlišuje také typy uživatelských účtů, což umožňuje systém nasadit v prostředí, kde jsou služby pronajímány a uživatelé k nim mají pouze částečný přístup.

Současně s implementací systému byla rozšířena MIB o nový modul CIFS-MIB definující monitorované data u služby typu fileserver. Tento modul je použitelný i v jiných systémech pro správu využívajících protokol SNMP.

Celý systém je distribuován ve virtuálním prostředí, kde je možné si jej před nasazením

na server vyzkoušet, nebo jej rovnou využít i s dodávanou instalací síťového serveru.

V budoucnu by implementace mohla být rozšířena například o možnost akumulovat statistiky z více nezávislých serverů, dále je zpracovávat a pod. Přínosem by mohla být i implementace propracovanější správy uživatelů, kdy by každý uživatel vlastnil konkrétní statistiku a mohl s ní podle toho manipulovat, nebo implementace automatického zasílání zpráv při kritické hodnotě některé statistiky.

Seznam použitých zdrojů

[1] *Manuálová stránka operačních systémů typu UNIX, automatický spouštěč aplikací CRONd.* [Online; navštíveno 5.5.2007].
URL http://www.die.net/doc/linux/man/man8/crond.8.html

[2] *Manuálová stránka operačních systémů typu UNIX, příkazový řádek BASH.* [Online; navštíveno 1.4.2007].
URL http://www.die.net/doc/linux/man/man8/crond.8.html

[3] *Manuálová stránka operačních systémů typu UNIX, syntaxe konfiguračního souboru aplikace SNMPd.* [Online; navštíveno 4.4.2007].
URL http://www.die.net/doc/linux/man/man5/snmpd.conf.5.html

[4] *Online vyhledání v Management Information Base (MIB).* [Online; navštíveno 5.5.2007].
URL http://www.mibsearch.com/

[5] *Stránky balíku aplikací NET-SNMP.* [Online; navštíveno 15.4.2007].
URL http://net-snmp.sourceforge.net/docs/man/

[6] *Stránky OpenSource databáze MySQL.* [Online; navštíveno 28.4.2007].
URL http://www.mysql.com

[7] *Stránky PHP knihovny pro generování grafů JpGraph.* [Online; navštíveno 10.5.2007].
URL http://www.aditus.nu/jpgraph/

[8] *Stránky skriptovacího jazyka PHP Hypertext Preprocesor.* [Online; navštíveno 1.5.2007].
URL http://www.php.net

[9] *Stránky webserveru Apache.* [Online; navštíveno 28.4.2007].
URL http://www.apache.org

[10] Case, J.; Fedor, M.; Schoffstall, M.; aj.: Simple Network Management Protocol (SNMP). RFC 1157 (Historic), Květen 1990, [Online; navštíveno 19.4.2007].
URL http://www.ietf.org/rfc/rfc1157.txt

[11] Fajmon, I., B. Růžičková: *Matematika 3.* Brno: Skriptum FEKT VUT, 2005, 147–167 s.

[12] Kay, T.: *Linux+ Cerfification Bible.* Hungry Minds, 2002, ISBN 0-7645-4881-6.

[13] McCloghrie, K.; Rose, M.: Management Information Base for network management of TCP/IP-based internets. RFC 1156 (Historic), Květen 1990, [Online; navštíveno 20.4.2007].
URL http://www.ietf.org/rfc/rfc1156.txt

[14] Molle, M. L.: Binary Logarithmic Arbitration Method for Ethernet. Technická zpráva, University of Toronto, Canada, Duben 1994, [Online; navštíveno 8.5.2007].
URL http://www.ethermanage.com/ethernet/pdf/molle-report.pdf

Seznam použitých zkratek a symbolů

WWW – World Wide Web

HTML – HyperText Markup Language

XHTML – eXtensible HyperText Markup Language

CSS – Cascading Style Sheet

HTTP – HyperText Transfer Protocol

HTTPS – HTTP Over SSL

FTP – File Transfer Protocol

CIFS – Common Internet FileSystem

SMB – Server Message Block

SNMP – Simple Network Management Protocol

OS – Operation System

VM – Virtual Machine

NAT – Network Address Translation

GPL – General Public Licence

RFC – Request For Comment

MTU – Maximum Transmission Unit

MIB – Management Information Base

PHP – PHP Hypertext Preprocessor

SQL – Structured Query Language

OID – Object IDentifier

CPU – Central Processing Unit

ER Diagram – Entity-Relationship Diagram

BASH – Bourne-Again SHell

DVD – Digital Versatile (Video) Disc

Seznam příloh

Příloha A

Definice nové MIB za použití normy ASN.1

```
-- *-----------------------------------------------------------------
-- *
-- *           CIFS-MIB.my:  CIFS statistics MIB.
-- *
-- *  experimental purposes, part of Bachelor's thesis
-- *
-- *  Petr Pospichal 2007
-- *
-- *-----------------------------------------------------------------

CIFS-MIB DEFINITIONS ::= BEGIN

IMPORTS
netSnmpExamples               FROM NET-SNMP-EXAMPLES-MIB
OBJECT-TYPE, Counter32, Integer32 ,
MODULE-IDENTITY                       FROM SNMPv2-SMI
MODULE-COMPLIANCE, OBJECT-GROUP       FROM SNMPv2-CONF
  experimental FROM SNMPv2-SMI;

--
-- A brief description and update information about this mib.
--
cifsExperimentalMIB MODULE-IDENTITY
    LAST-UPDATED "200705071012Z"            -- 5.9.2007 10:12
    ORGANIZATION "FIT VUT Brno"
    CONTACT-INFO "
          Petr Pospichal
          E-mail: xpospi45@stud.fit.vutbr.cz"

    DESCRIPTION  "
    The CIFS Statistics MIB models counters and objects
    that are of management interest of CIFS

          Acronyms
          The following acronyms are used in this document:

            CIFS:       Common Internet File System

            MIB:        Management Information Base
         "
    ::= { experimental 255 }


cifsAgentModules   OBJECT IDENTIFIER ::= { cifsExperimentalMIB 1 }
```

```
--cifsAgentModules

cifsSummaryOutBytes OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
       "Total amount of bytes sent by CIFS server."
::= { cifsAgentModules 1 }

cifsSummaryOutPkts OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
       "Total amount of packets sent by CIFS server."
::= { cifsAgentModules 2 }

cifsSummaryInBytes OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
       "Total amount of bytes received by CIFS server."
::= { cifsAgentModules 3 }

cifsSummaryInPkts OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
       "Total amount of packets received CIFS server."
::= { cifsAgentModules 4 }


cifsInFiles OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
       "Total amount of files received by CIFS server."
::= { cifsAgentModules 5 }

cifsOutFiles OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
       "Total amount of files sent by CIFS server."
::= { cifsAgentModules 6 }

cifsPerfCPULoad OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
       "CPU utilization caused by CIFS server."
::= { cifsAgentModules 7 }


END
```

Příloha B

Použité SNMP objekty a jejich OID

jméno	OID	modul
wfHttpSummaryOutBytes	.1.3.6.1.4.1.18.3.5.3.22.1.2.1.9	WF-HTTP-MIB
wfHttpSummaryResponses	.1.3.6.1.4.1.18.3.5.3.22.1.2.1.4	WF-HTTP-MIB
cceHttpPerfCpuLoad	.1.3.6.1.4.1.9.9.178.1.1.2.8	CISCO-CONTENT-ENGINE-MIB
alHttpStatsActiveSessions	.1.3.6.1.4.1.3076.2.1.2.7.1.7	ALTIGA-HTTP-STATS-MIB
alHttpStatsActiveConnections	.1.3.6.1.4.1.3076.2.1.2.7.1.5	ALTIGA-HTTP-STATS-MIB

Tabulka B.1: přehled použitých MIB objektů pro službu WWW

jméno	OID	modul
wfFtpInFiles	.1.3.6.1.4.1.18.3.5.3.10.1.15	Wellfleet-FTP-MIB
wfFtpOutFiles	.1.3.6.1.4.1.18.3.5.3.10.1.16	Wellfleet-FTP-MIB
wfFtpLogins	.1.3.6.1.4.1.18.3.5.3.10.1.11	Wellfleet-FTP-MIB
alFtpClientStatsOctetsXmit	.1.3.6.1.4.1.3076.2.1.2.12.2.11	ALTIGA-FTP-STATS-MIB
alFtpClientStatsOctetsRecv	.1.3.6.1.4.1.3076.2.1.2.12.2.12	ALTIGA-FTP-STATS-MIB

Tabulka B.2: přehled použitých MIB objektů pro službu FTP

jméno	OID	modul
ifOutOctets	.1.3.6.1.2.1.2.2.1.16	IF-MIB
ifInOctets	.1.3.6.1.2.1.2.2.1.10	IF-MIB
ifInDiscards	.1.3.6.1.2.1.2.2.1.13	IF-MIB
ifInErrors	.1.3.6.1.2.1.2.2.1.14	IF-MIB
ipForwDatagrams	.1.3.6.1.2.1.4.6	IP-MIB
nwIpFwdIfCtrFwdBytes	.1.3.6.1.4.1.52.4.2.2.2.3.1.2.2.2.2.1.1.15	CTRON-IP-ROUTER-MIB

Tabulka B.3: přehled použitých MIB objektů pro síťová rozhraní

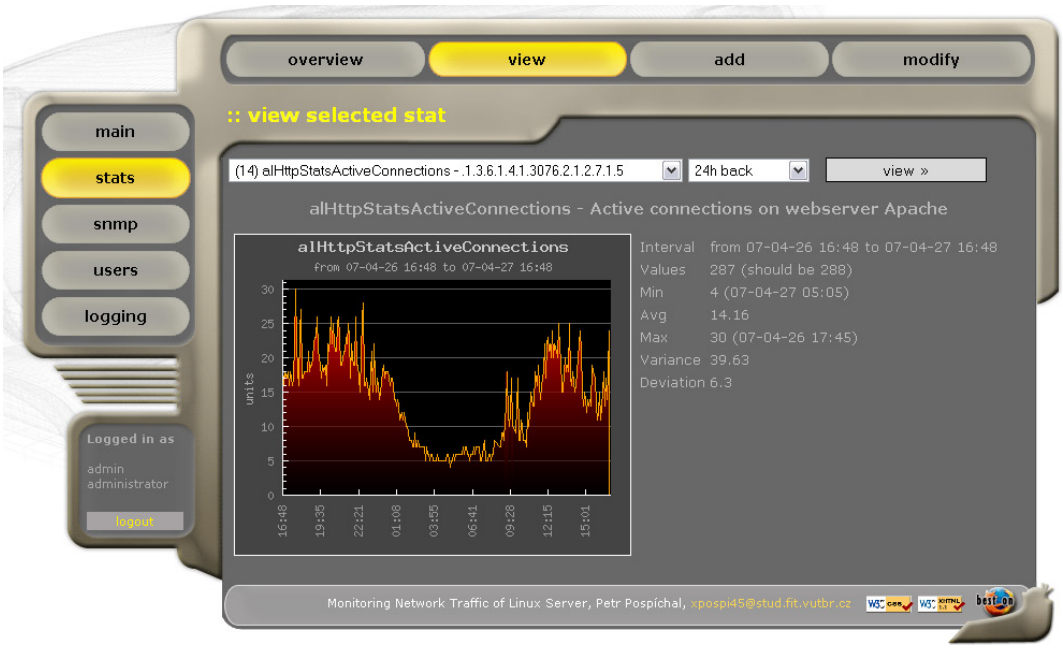
jméno	OID	modul
CIFSSummaryOutBytes	.1.3.6.1.3.255.1.1	CIFS-MIB
CIFSSummaryOutPkts	.1.3.6.1.3.255.1.2	CIFS-MIB
CIFSSummaryInBytes	.1.3.6.1.3.255.1.3	CIFS-MIB
CIFSSummaryInPkts	.1.3.6.1.3.255.1.4	CIFS-MIB
CIFSInFiles	.1.3.6.1.3.255.1.5	CIFS-MIB
CIFSOutFiles	.1.3.6.1.3.255.1.6	CIFS-MIB
CIFSPerfCPULoad	.1.3.6.1.3.255.1.7	CIFS-MIB

Tabulka B.4: definice nových MIB objektů pro CIFS

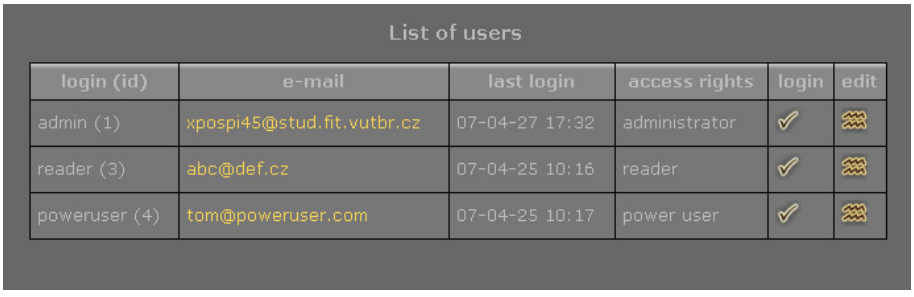
Použité OID reprezentují základní objekt, v případě tabulárních objektů konkrétní entita obsahuje ještě číslo instance.

Příloha C

Ukázky vzhledu webového rozhraní

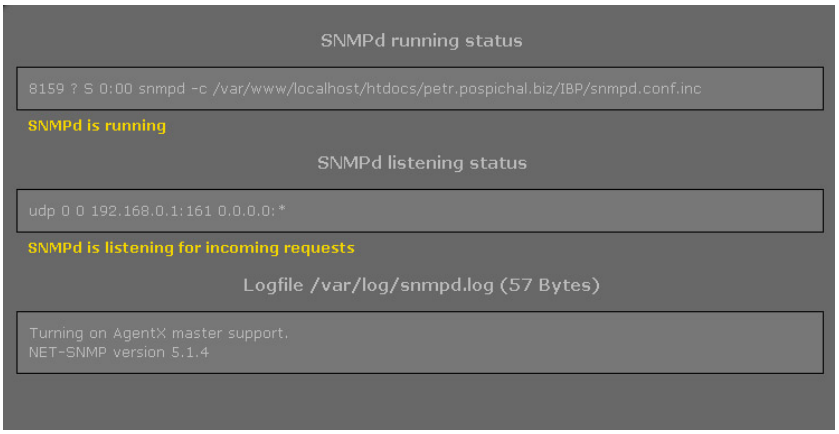


Obrázek C.1: ukázka celkového vzhledu rozhraní s dlouhodobějším vyhodnocením statistiky

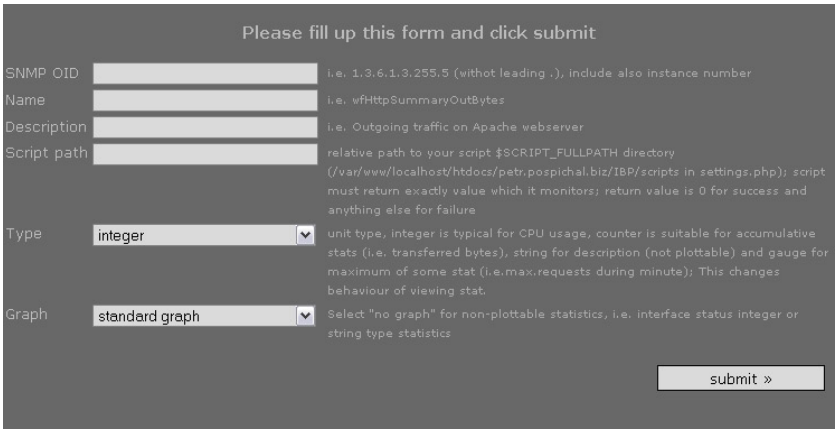


Obrázek C.2: výřez rozhraní zobrazujícího seznam uživatelů

Obrázek C.3: výřez rozhraní zobrazujícího výpis z logu



Obrázek C.4: výřez rozhraní zobrazujícího stav SNMP aplikace



Obrázek C.5: výřez rozhraní zobrazujícího formulář pro přidání statistiky