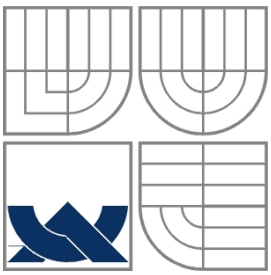
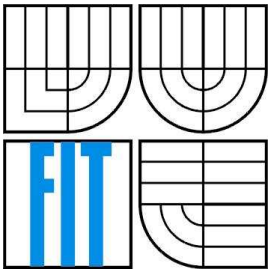


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

WEBOVÝ PORTÁL PRO SLEDOVÁNÍ ODEZEV SÍTĚ A SLUŽEB

WEB PORTAL FOR NETWORK AND APPLICATION RESPONSE TIME MONITORING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — PAVOL HORNICKÝ
AUTHOR

VEDOUCÍ PRÁCE — Ing. JIŘÍ TOBOLA
SUPERVISOR

BRNO 2009

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů Akademický rok 2008/2009

Zadání bakalářské práce

Řešitel: **Hornický Pavol**
Obor: Informační technologie
Téma: **Webový portál pro sledování odezev sítě a služeb**
Kategorie: Web

Pokyny:

1. Seznamte se s technologiemi pro tvorbu webových informačních systémů (HTML, CSS, PHP, Javascript, MySQL apod.).
2. Prostudujte možnosti monitorování odezev sítě, síťových prvků a aplikací.
3. Proveďte analýzu požadavků pro systém umožňující sledování a přehledné zobrazení časových odezev sítě, serverů a služeb. Systém musí poskytovat podporu monitorování široké škály služeb a uživateli musí nabídnout přehledné webové rozhraní s přehledem výsledků a nastavením monitorování.
4. Vytvořte detailní návrh tohoto systému a vhodně jej modelujte.
5. Navržený systém realizujte a otestujte, funkčnost systému demonstrujte na vhodně zvoleném vzorku dat.
6. Zhodnoťte dosažené výsledky a diskutujte možnosti dalšího rozšíření systému.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění prvních tří bodů zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Tobola Jiří, Ing.**, UPSY FIT VUT
Datum zadání: 1. listopadu 2008
Datum odevzdání: 20. května 2009

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
612 66 Brno, Božetěchova 2

L.S.

doc. Ing. Zdeněk Kotásek, CSc.
vedoucí ústavu

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací systému pro monitorování časových odezev síťových služeb. Práce popisuje základní techniky využívané při takovém monitorování a podrobně popisuje aktivní monitorování. Jádrem výsledného systému je automatizované měření odezev síťových služeb pomocí standardních unixových programů. Naměřené odezvy jsou pomocí rrdtool zobrazované v přehledných grafech přes webové rozhraní, které se stará i o správu monitorování.

Abstract

This bachelor's thesis describes the design and implementation of system for the response time monitoring of network services. This thesis describes the basic techniques used for such monitoring, and describes in detail the active monitoring. The core of the resulting system is the automated responses measurement of network services using standard UNIX programs. The web interface gives a clear view of captured responses using rrdtool graphs. The web interface is also used for monitoring management.

Klíčová slova

monitorování síťových služeb, aktivní monitorování, pasivní monitorování, rrdtool, cron

Keywords

Monitoring of network services, active monitoring, passive monitoring, rrdtool, cron

Citace

Pavol Hornický: Webový portál pro sledování odezev sítě a služeb, bakalářská práce, Brno, FIT VUT v Brně, 2009

Webový portál pro sledování odezev sítě a služeb

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jiřího Toboly. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Pavol Hornický
7.5.2009

Poděkování

Rád by som poďakoval vedúcemu práce Ing. Jiřímu Tobolovi za odbornú pomoc pri riešení práce.

Obsah

1 Úvod

Funkčnosť niektorých aplikácií môže byť rozhodujúca pre každodenný chod, a tiež dlhodobú prosperitu väčšiny organizácií. Ide hlavne o aplikácie a služby, ktoré úzko súvisia s interakciou voči zákazníkovi ale aj nástroje často využívané zamestnancami alebo študentmi. Infraštruktúry týchto aplikácií a zariadení, ako napr. SQL server alebo dynamické webové aplikácie je potrebné mať k dispozícii dvadsaťštyri hodín denne, sedem dní v týždni. Ich výpadky môžu mať v konečnom dôsledku veľmi nepríjemné následky. K tomu aby sme sa o týchto výpadkoch včas dozvedeli, a mohli lepšie skúmať ich príčiny nám slúžia monitorovacie nástroje.

Pretože väčšina organizácií monitorovania svojich systémov stále podceňuje a nevenuje im dostatočnú pozornosť. Veľké množstvo výpadkov týchto systémov a užívateľských problémov spôsobených nefunkčnosťou, alebo inou chybou systému, je dnes však stále zisťovaných práve zákazníkmi a koncovými užívateľmi a nie IT. Najvhodnejšie by bezpochyby bolo aby, ak už problém nastane, bol čo najrýchlejšie odstránený a zákazník sa s ním nestretol. I situácia, keď zákazník odhalí problém, nahlási nám ho a my ho ubezpečíme, že je nám daný problém i jeho príčina známa a tiež že vieme ako a kedy ho vyriešime pôsobí oveľa prívetivejšie, ako keď sa o problémoch dozvedáme od samotných zákazníkov. I preto má monitorovanie a kvalitné monitorovacie nástroje pre nás kľúčový význam.

Ďalšou veľkou výhodou, ktorú nám kvalitný monitorovací nástroj poskytuje, je aj urýchlenie riešenia problému. Odhaduje sa, že asi 80% času stráveného riešením problému spočíva v samotnom nájdení presného miesta kde problém vznikol a tiež príčiny vzniku tohto problému. Monitorovací nástroj, ktorý systematicky kontroluje beh celého systému a poprípade nás i priebežne o stave informuje môže teda značne toto hľadanie urýchliť, ušetriť nám množstvo času a celkovo skrátiť dobu výpadkov alebo nefunkčnosti. Toto šetrenie času má veľmi veľký význam, napríklad ak sú niektoré služby závislé na iných a chyba jednej z nich spôsobí i nedostupnosť ostatných. Alebo tiež ak sú rozdelené na viacero zdrojov a miest, dokonca ak sú fyzicky rozdelené napr. na viacerých serveroch, ktoré sa môžu nachádzať na rozdielnych miestach, čiže na odstránenie problému je potrebný presun medzi nimi a my by sme v takejto situácii nevedeli na ktorom zo serverov chyba nastala, museli ich postupne kontrolovať všetky. Čas na odstránenie problému by v takejto situácii rapídne vzrástol. Dlhodobejšie porušená úroveň služieb môže v praxi spôsobiť napr. finančnú ujmu, alebo narušenie zákazníckych vzťahov.

Témou tejto práce je teda oboznámiť s možnosťami monitorovania odoziev siete, sieťových aplikácií a služieb, analyzovať požiadavky na vytvorenie systému, ktorý by toto sledovanie vykonával, a tiež detailný návrh a implementácia takéhoto systému.

Práca v úvodnej časti popisuje základné techniky, ktoré sa pri monitorovaní využívajú, porovnáva ich, stručne popisuje ich výhody a nevýhody a podrobnejšie rozoberá aktívne monitorovanie. Ďalšia kapitola zhrňuje požiadavky na daný systém. Zaoberá sa automatizovaným spúšťaním monitorovacích skriptov, možnosťami zobrazenia nameraných výsledkov a návrhom RRD a SQL databázy. Obsahuje tiež návrh webového rozhrania pre správu monitorovaní. V predposlednej kapitole nájdeme popis vlastnej implementácie systému. Záver zhodnocuje vytvorený systém a opisuje jeho nedostatky a možné rozšírenia. Popísané sú tu tiež namerané hodnoty, presnosť a efektívnosť riešenia.

2 Techniky monitorovania sieťových aplikácií a služieb

2.1 Základné techniky monitorovania

Už pri vzniku internetu boli vytvorené základné postupy na zaistenie monitorovania siete. Boli to napr. protokoly ICMP a neskôr tiež SNMP, ktoré sa vo veľkej miere používajú dodnes.

2.1.1 ICMP – Internet Control Message Protocol

Protokol popísaný štandardom RFC 792 je súčasťou IP vrstvy modelu TCP/IP. Internetový protokol nie je navrhnutý tak, aby mohol byť absolútne spoľahlivý. ICMP slúži k odovzdávaniu riadiacich správ o problémoch IP komunikácie. Účelom týchto riadiacich správ nie je dosiahnutie stopercentnej spoľahlivosti IP, ale informovať o problémoch v komunikačnom prostredí. Stále nie je zaručené, že datagram bude doručený, alebo že bude vrátená kontrolná správa. Aby nedochádzalo k nekonečnému zacyklovaniu správ o správach a podobne, ICMP správy obvykle informujú o chybách pri spracovávaní datagramov [5].

ICMP definuje 11 typov správ: Echo Reply, Destination Unreachable, Source Quench, Redirect, Echo, Time Exceeded, Parameter Problem, Timestamp, Timestamp Reply, Information Request, Information Reply. ICMP protokol využívajú aj niektoré nástroje na diagnostiku sietí ako napr. PING, či TRACEROUTE.

2.1.2 SNMP – Simple Network Management Protocol

Popisuje ho štandard RFC 1157 [6].Tento protokol definuje dva druhy zariadení, a to - sledované zariadenia (managed devices), ktoré uchovávajú štatistické údaje o svojej činnosti, a centrálnu riadiacu stanicu NMS (network management station), ktorá získava informácie o sledovaných zariadeniach analyzuje ich a zobrazuje administrátorovi. Protokol je tvorený len jednoduchými príkazmi typu načítaj: dáta (get, get-next), zapíš dáta (set), pošli správu (trap, notification), a ešte niekoľkými riadiacimi príkazmi (inform, report). Na architektúre SNMP je zaujímavá databáza dát obsahujúca informácie o sieťových prenosoch, konfigurácii zariadení a pod. Táto databáza sa nazýva MIB (Management Information Base). MIB definuje typy sledovaných dát v danom zariadení a ich jednoznačnú identifikáciu pomocou hodnoty OID (object identifier). Monitorovanie

siete pomocou SNMP spočíva v zisťovaní vybraných hodnôt SNMP zariadení a ich následnom spracovaní.

2.1.3 Aktívne a pasívne monitorovanie

ICMP a SNMP protokoly tvoria len základ techník využívaných na sledovanie sietí. Dnes už existuje veľké množstvo komplexnejších riešení na sledovanie sietí a sieťových služieb, ktoré sú schopné testovať nielen to, či sú služby k dispozícii, ale aj kontrolovať ich stav, tj. sledovať, ako spoľahlivo pracujú, aké sú ich odozvy a pod. Pri týchto sledovaniach sa využívajú dve základné techniky: **aktívne a pasívne monitorovanie**.

2.2 Porovnanie aktívneho a pasívneho monitorovania

V tejto kapitole si popíšeme základné princípy aktívneho a pasívneho prístupu k monitorovaniu sieťových služieb. Tiež budú spomenuté hlavné výhody a nevýhody oboch týchto prístupov a navrhnuté ideálne riešenie .

2.2.1 Aktívne monitorovanie (Active Monitoring)

alebo tiež „ Synthetic monitoring“ (syntetické sledovanie) používa metodiku, ktorá sa snaží simulovať chovanie skutočných užívateľov v sieti. Znamená to, že pri aktívnom sledovaní sa na danú službu posielajú dotazy, pri ktorých sa predpokladá, že by ich chcel robiť skutočný užívateľ. Je preto veľmi dôležité, pozorne navrhnúť spôsob sledovania, aby bol čo najreprezentatívnejší vzhľadom k predpokladaným reálnym transakciám. Týmto spôsobom jednoducho zistíme, ako by sa v danom okamihu služba zachovala, a aká by bola doba jej odozvy. Nezistíme však skutočnú situáciu a skutočné správanie pri reálnych užívateľských aktivitách. Aktívne sledovanie siete je preto vhodné na predvídanie rôznych situácií, čím môžeme predchádzať možným problémom. Táto skutočnosť je asi najväčšou výhodou aktívneho monitorovania.

Problémom aktívneho prístupu k monitorovaniu systému môže byť napríklad oddelenie simulovaných transakcií nad systémom od skutočných užívateľských transakcií. Napríklad ak by syntetická transakcia mala simulovať zápis do databázy SQL, aby sme zistili ako rýchlo je SQL server schopný zápis zrealizovať, musí byť táto transakcia pozorne navrhnutá tak, aby sa nám testovacie dáta a skutočné užívateľské dáta nepomiešali. Na zmiernenie týchto potenciálnych problémov by bolo najvhodnejšie zriadiť špecializované testovacie účty, aby bolo jednoduchšie

rozhodnúť, či žiadosť prišla od reálneho užívateľa, alebo ide iba o syntetický test. Pri operáciách, ktoré zahrňujú zmenu údajov je vhodné určiť spôsoby ako vylúčiť tieto údaje z prehľadov dát a pokiaľ je to možné, hľadať spôsoby ako odstrániť testovacie dáta zo systému. Toto nemusí byť vždy jednoduchá úloha.

Nevýhodou aktívneho (syntetického) prístupu k monitorovaniu je tiež to, že syntetické sledovanie má síce potenciál k napodobňovaniu všetkých možných scenárov, ale správcovia systému nemôžu predpokladať všetky situácie a vôbec nie ich všetky simulovať, pretože by to bolo nákladné i na čas i na prostriedky.

2.2.2 Pasívne monitorovanie (Passive Monitoring)

Pasívne alebo tiež real-time monitorovanie skúma skutočné užívateľské transakcie s cieľom odhaliť a zachytiť chyby a spomalenia systému. Pasívne monitorovanie je väčšinou založené na hardwarovom vybavení, ktoré je vo vnútri systému a zachytáva prevádzku.

Network monitoring interface card alebo tiež NMIC je podobná ako sieťová karta (NIC). Na rozdiel od štandardnej sieťovej karty je NMIC navrhnutá tak, aby pasívne naslúchala na sieti. Na funkčnej úrovni sa môže NMIN od NIC líšiť v tom že NMIC nemusí mať MAC adresu, môže jej tiež chýbať schopnosť odosielať dáta, a nemusí byť v sieti vôbec viditeľná. NMIC sa obvykle používajú na prietokové analýzy, monitorovanie siete a tiež ako protokolové analyzátory systému.

Pasívne sledovanie môže byť veľmi užitočné pri riešení problémov s výkonom, hneď ako sa tieto problémy objavia. Líši sa od aktívneho sledovania v tom, že sa opiera o skutočnú prevádzku, takže problémy môžu byť odhalené až po ich vzniku.

2.2.3 Najvhodnejší prístup

Zo začiatku bolo pasívne monitorovanie chápané ako konkurenčná technika k syntetickému prístupu. V súčasnosti už väčšina sieťových odborníkov uznáva, že tieto dva prístupy sa vzájomne vhodne dopĺňajú a ich kombináciou môžu vzniknúť naozaj kvalitné systémy na sledovanie a správu sietí a sieťových služieb a aplikácií.

2.2.4 Aktívne monitorovanie

Ako už bolo spomenuté, aktívne monitorovanie používa metodiky, ktoré sa snažia simulovať chovanie skutočných užívateľov v sieti. Podľa komplexnosti týchto simulácií a kvality ich návrhu môžeme tento spôsob monitorovania rozdeliť na niekoľko úrovní.

2.2.5 Úroveň socketov

Prvou a najjednoduchšou úrovňou je sledovanie na úrovni socketov. Týmto spôsobom rýchlo a efektívne simulujeme snahu o pripojenie sa k danej službe na našom serveri . Ak daná služba na serveri nie je nainštalovaná, alebo nepracuje, tak už pri pripájaní sa na daný port dostaneme chybu. Ak služba funguje, môžeme odmerať ako dlho pripájanie k danému portu trvá, a tým vlastne dosiahneme základné monitorovanie danej služby .

Toto monitorovanie nám však poskytuje iba informácie o tom, či služba na danom serveri funguje, teda či je dostupná a či sa na daný port dá pripojiť. Neposkytuje nám ale informácie o tom, ako rýchlo je služba schopná odpovedať na naše reálne dotazy, ktoré by skutočný užívateľ zadával.

2.2.6 Úroveň komunikácie so službou

Ďalšou úrovňou sledovania je sledovanie, kedy sa na server nielen pripojíme a testujeme, či server danú službu podporuje, ale i testovanie, ako rýchlo je daná služba schopná odpovedať na naše reálne dotazy .

To znamená, že pri tejto úrovni sa rovnako ako v predchádzajúcom prípade pripojíme na server a po úspešnom pripojení začneme serveru posielať reálne dotazy. Napríklad, ak chceme sledovať port číslo 80, na ktorom beží http protokol, pošleme serveru príkaz GET, a zmeriame ako rýchlo nám server danú požiadavku spracuje a odpovie.

Týmto spôsobom simulujeme užívateľské príkazy i na ostatné porty a služby. Toto riešenie je oproti prvej úrovni oveľa kvalitnejšie, spoľahlivejšie, a zároveň nám pomáha odhaľovať veľa problémov, na ktoré by sme pri prvom spôsobe vôbec neprišli. Ide napríklad o to, keď je daná služba síce na serveri spustená a dalo by sa k nej pripojiť, ale nepracuje správne, alebo by jej spracovanie konkrétnych dotazov trvalo z nejakého dôvodu príliš dlho . Tiež sa môže stať, že na danom porte bude spustená úplne iná služba ako je dohodnuté. Čiže napríklad na porte číslo 25, ktorý je typický pre SMTP - Simple Mail Transfer Protocol, teda pre prenos správ elektronickej pošty, bude spustený HTTP alebo FTP, ktorých príkazy sú rozdielne.

2.2.7 Úroveň kompletnej simulácie užívateľských aktivít

Najkomplikovanejším a najkomplexnejším spôsobom monitorovania je monitorovanie, pri ktorom máme navrhnuté, nielen pripojenie k serveru na daný port a kontrolu, či daná služba na porte funguje správne jednoduchými príkazmi, ale aj simulovanie komplikovanejších užívateľských príkazov a operácii, ako sú napr. prihlasovanie sa k FTP servu a zápis dát na server, alebo tiež napr. zápis dát do databázy SQL.

Pri tomto spôsobe monitorovania však vznikajú problémy, ktoré boli už spomenuté pri základnej charakteristike aktívneho prístupu k monitorovaniu. Ide o problémy s odlišovaním dát použitých pri simulácii od skutočných užívateľských dát, zaťažovaním systému testami a pod. Preto sú síce monitorovacie nástroje využívajúce tento spôsob najkomplexnejšie, ale nemusia byť vždy i najvhodnejšie.

2.2.8 Iné kritériá

Rozdelenie spôsobov aktívneho sledovania na tieto úrovne podľa komplexnosti však nie je jediným kritériom, ktorým sa dá posudzovať kvalita testov. Na celkovú kvalitu sledovania má vplyv ešte množstvo vecí. Keďže vo väčšine prípadov chceme mať dlhodobý prehľad o stave nášho systému a jeho efektivite, je veľmi dôležitý i celkový návrh testov. Pri tomto návrhu sa zohľadňujú kritériá, ako sú napríklad možnosti nastavenia intervalu testovania, plánovania monitoringu a spracovania získaných dát z testov.

3 Návrh systému

V tejto kapitole je popísaný návrh systému, požiadavky na systém a tiež špecifikácia niektorých sieťových služieb a protokolov, s ktorými systém pracuje.

3.1 Analýza požiadaviek

Systém má umožňovať monitorovanie širokej škály sieťových služieb, a tiež prehľadné zobrazenie časových odoziev pomocou webového rozhrania. Základné otázky teda sú: Ako merať odozvy sieťových služieb? Ako prehľadne zobraziť výsledky užívateľovi cez webové rozhranie?

Na prvú z týchto otázok si vieme jednoducho odpovedať. Program, ktorý sa pripojí cez sieť na danú službu, pošle službe dotaz a zmeria čas, za aký je služba schopná odpovedať nie je problémové naimplementovať. Otázne zostáva, či je toto riešenie pre túto prácu najvhodnejšie, keďže musí systém poskytovať monitorovanie širokej škály sieťových služieb.

3.1.1 Simulácia užívateľských akcií

V systéme máme simulovať užívateľské akcie, ktoré by užívatelia robili za bežných okolností nejakým programom. Opäť použijeme jednoduchý príklad s http protokolom: Chceme simulovať užívateľské akcie voči webovému serveru. Užívateľ najčastejšie zadáva do prehliadača stránku, ktorú chce zobraziť, teda chce aby mu ju server poslal. Okrem klasických grafických prehliadačov, ktoré sú v dnešnej dobe najrozšírenejšie existuje tiež mnoho nástrojov, ktoré dokážu serveru tento dotaz poslať. Napr. unixový program Wget, ktorý je súčasťou väčšiny unixových distribúcií.

Preto som najprv naimplementoval program, ktorý sa pomocou socket pripojí na daný port na serveri, pošle mu požiadavku a po úspešnej odpovedi sa ukončí. Potom som porovnal časy namerané týmto programom s časmi nameranými pomocou programu Wget, ktorý okrem pripojenia a poslania, požiadavky stránku tiež zobrazí a uloží do súboru. Vo výsledku bolo viditeľné, že tieto časy sa líšia len minimálne. Takto som dospel k záveru, že implementovať podobný program na všetky služby, ktoré má systém podporovať by bolo pre túto prácu zbytočné a neefektívne. Na pripájanie a komunikáciu so servermi a službami je vhodným riešením použitie štandardných programov.

3.1.2 Zobrazovanie nameraných hodnôt

Pokúsime sa teraz zodpovedať aj druhú otázku: Ako dosiahnuť prehľadné zobrazenie výsledkov užívateľovi pomocou webového rozhrania? Ku koncu tohto oddielu potom vyvodíme z týchto odpovedí požiadavky, ktoré systém musí spĺňať.

Na prehľadné zobrazenie nameraných výsledkov je možné použiť dve možnosti. Tou jednoduchšou, je ukladanie výsledkov do tabuliek a ich vypisovanie na stránku. Toto riešenie môže byť prehľadné, ale málo ilustratívne. Z tabuľky plnej čísel si totiž užívateľ, ktorý si tieto výsledky prezerá, ťažko predstaví, kedy mal server výpadok, alebo kedy boli odozvy väčšie. Aj keď by sa pre lepšiu ilustratívnosť dali tieto výsledky farebne odlíšiť alebo inak zvýrazniť, oveľa vhodnejším riešením bude v tomto prípade zobrazovanie nameraných hodnôt na stránke v grafoch. K tomuto účelu vhodne poslúžia „ The round-robin database tool – RRDtool“.

3.1.3 RRDtool

RRDtool [2] je výkonný Open-Source nástroj na spracovanie časových dát a na ich následné grafické vykresľovanie. Dáta sa cyklicky ukladajú do Round Robin databázy, čím je dosiahnuté, že spotreba času ani pamäte časom nenarastá. Tvorba grafického výstupu z takto uložených dát prebieha na základe požiadavky s možnosťou špecifikácie časového rozsahu a je od ich ukladania nezávislá.

3.1.4 Zhrnutie požiadaviek

Na implementáciu tohto systému budem teda potrebovať webový server, fungujúci na unixovom operačnom systéme, ktorý má povolené spúšťanie externých programov z php skriptov. Ďalej musia byť na tomto serveri nainštalované RRDtools. A keďže systém má bežať ako webový portál s možnosťou spravovania aktuálnych monitorovaní, a tiež prihlasovania užívateľov, databáza na tomto serveri bude tiež nevyhnutná.

3.2 Návrh

V tejto časti si popíšeme samotný návrh pozadia systému a webového rozhrania, a tiež návrh SQL databázy a rrd databázy pre vykresľovanie grafov.

3.2.1 Jadro systému + Cron

Aby sme dosiahli pravidelné meranie odoziev serverov, musí toto meranie prebiehať pravidelne na pozadí systému bez ohľadu na to či sú užívatelia prihlásení alebo nie. K tomuto nám poslúži softvérový démon Cron [1], ktorý na unixových systémoch automatizovane spúšťa nejaký proces. Môžeme ho tiež nazvať plánovačom úloh. Zapísaním príkazov do cron tabuľky, tzv. crontab si môžeme naplánovať pravidelné spúšťanie akéhokoľvek skriptu. Každý užívateľ v systéme ma svoj vlastný crontab. Do crontab sa príkazy ukladajú pomocou príkazu: `crontab -e`. Každá položka v súbore crontab začína piatimi stĺpcami, ktoré stanovujú okamžik spustenia príkazu.

`#MIN HOUR MDAY MON DOW COMMAND,` kde:

Položka	Definícia	Platné hodnoty
MIN	Minúta	0-60
HOUR	Hodina	0-23
MDAY	Deň Mesiac	1-31
MON	Mesiac	1-12 ALEBO jan, feb, mar, apr, …
DOW	Deň v týždni	0-6 ALEBO sun,mon,tue,wed,thu,fri,sat
COMMAND	Príkaz	Akýkoľvek platný príkaz

Tabuľka 3.1: Formátovanie crontab.

Úlohy cronu nieje vhodné spúšťať presne v celú hodinu, polhodinu alebo štvrťhodinu, ale niekoľko minút pred alebo potom. Znižuje sa tak pravdepodobnosť súčasného spúšťania viacero procesov, čo by mohlo krátkodobo príliš zaťažovať systém.

V crontab sú povolené tieto špeciálne znaky:

- Hviezdička: "*" – zastupuje každú položku v kategórii. (napr. každú minútu, každý deň)
- Čiarka: "," – slúži na oddeľovanie viacerých hodnôt v danej kategórii. (napr. 1,3,5)
- Pomlčka: "-" – špecifikuje rozsahy (napr. 10-20)
- Lomítko: "/" – určuje rozsah opakovania (napr. */5 – každých 5 minút)

Príklady:

`00 08 * * 1-5 script -` spustí script vždy v pondelok – piatok o 8:00

`*/5 8-20 * * * script` – spustí script každý deň, každých 5 minút medzi 8:00 a 20:00

`* * * * * script` – spustí script každú minútu každý deň

Príkaz podobný poslednému príkladu som použil aj pre úpravu crontabu v mojom systéme.

Týmto príkazom sa spúšťa každý deň každú minútu skript, ktorý obsahuje spúšťanie všetkých úloh potrebných na meranie odoziev a ukladanie týchto dát do rrd databázy. Pri odstránení alebo pridaní nového monitorovania sa obsah tohto skriptu automaticky mení podľa aktuálnej databázy sledovaní.

3.2.2 Webové rozhranie

Webové rozhranie bolo navrhnuté tak, aby po prihlásení užívateľovi poskytlo rýchlo pochopiteľné a intuitívne ovládanie systému.

Obrázok 3.1: Menu webového rozhrania systému.

Menu na obrázku 3.1 tvoria štyri základné oddiely: „Zobrazenie grafov“, „Správa užívateľov“, „Upozornenia“ a „Sledovania“. Posledný odkaz „Odhlásiť“ slúži na odhlásenie sa zo systému pri ukončení práce.

Po kliknutí na položku „**Zobrazenie grafov**“ v menu sa načíta stránka, na ktorej sa nachádzajú grafy odoziev pre všetky monitorované služby na všetkých serveroch. Vyhľadávanie konkrétneho grafu v tomto prehľade môže však byť pri väčšom počte sledovaní náročné. Užívateľ má preto možnosť výberu serveru, služieb a časového úseku zobrazovaných grafov.

V sekcii „**Správa užívateľov**“ sa nachádzajú:

- Formulár na registráciu nového užívateľa
- Prehľad registrovaných užívateľov – tento prehľad obsahuje meno, priezvisko a informáciu, či je daný užívateľ administrátorom.

Sekciu „**Upozornenia**“ slúži na správu mailových upozornení v prípade výpadku, alebo nedostupnosti niektorého zo sledovaných serverov. Túto sekciu tvoria:

- Formulár na pridávanie mailovej adresy pre posielanie upozornení – vypĺňajú sa tu:
  - e-mail – mailová adresa kam sa mail pošle
  - služba – výber služby, pri ktorej výpadku sa má mail posielať
  - odozva – určuje odozvu ktorá sa bude považovať za výpadok
  - čas nedostupnosti – určuje časový interval v ktorom ak sú všetky odozvy serveru väčšie ako odozva považovaná za výpadok, odošle sa e-mailové upozornenie.
- Prehľad emailových adries na ktoré sa posielajú upozornenia

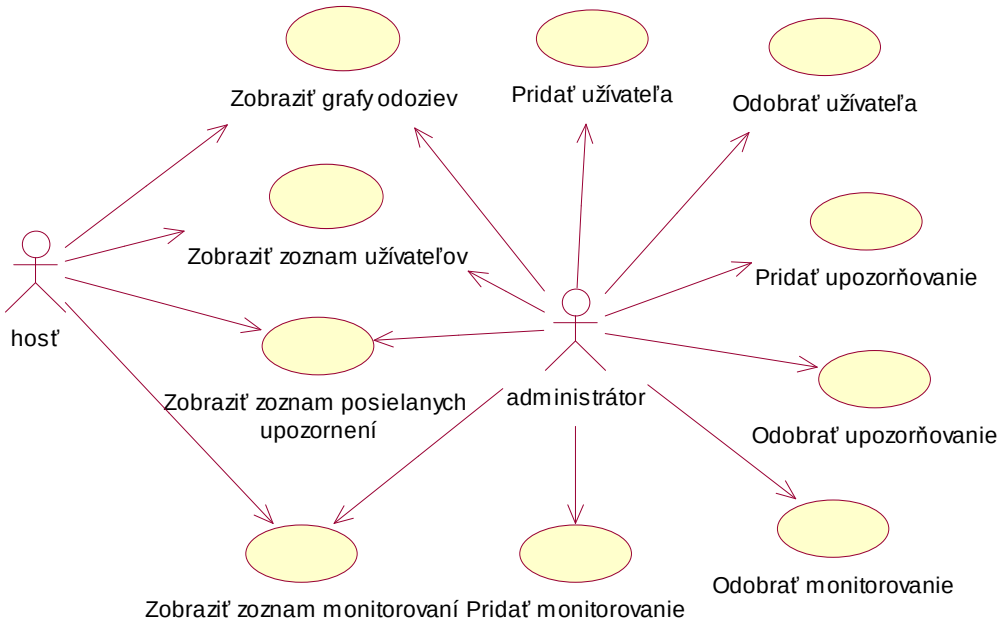
Poslednou sekciou je sekcia „**Sledovania**“ . Tu sa nachádzajú:

- Tabuľka so stručným prehľadom aktuálnych monitorovaní. Nad tabuľkou má užívateľ možnosť nastaviť, ako sa budú zobrazované dáta radiť (podľa názvu servera, názvu monitorovanej služby alebo dátumu pridania monitorovania do systému). Prehľad

obsahuje i možnosť monitorovanie zo systému odstrániť, ktorá je prístupná iba administrátorom systému.

- Interaktívny formulár pre pridanie nového monitorovania. Tento formulár sa pri výbere služby prispôsobí, a doplnia sa políčka, ktoré sú potrebné na správne sledovanie danej služby. Formulár tiež obsahuje tlačidlo test slúžiace na prekontrolovanie zadaných údajov a zistenie, či je možné konkrétnu službu monitorovať. Po úspešnej kontrole sa zobrazí tlačidlo pridať, pomocou ktorého sa údaje pridajú do databázy sledovaní a vytvorí sa Round Robin databáza na ukladanie a zobrazovanie nameraných hodnôt.

V systéme môžu vystupovať dva druhy užívateľov: bežný užívateľ (hosť) a administrátor (admin). Ich úlohu a možnosti zobrazuje diagram prípadov užitia na obrázku 3.2.



Obr. 3.2: Use Case Diagram systému

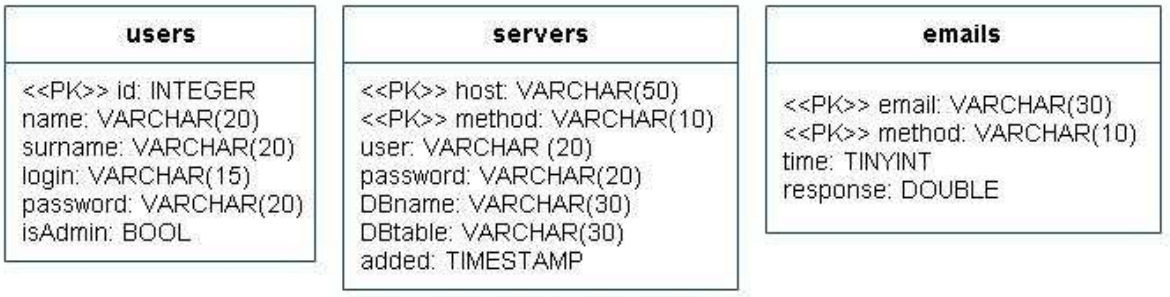
3.2.3 SQL Databáza

Pri návrhu databázy je dôležité ujasniť si, aké dáta potrebujeme ukladať, a aké hodnoty budú nadobúdať. My budeme monitorovať odozvy siete na rôznych serveroch. Na každom z nich je možné sledovať viacero služieb. Celkový návrh tabuliek systému zobrazuje ER diagram na obrázku 3.3. Na to, aby toto sledovanie prebiehalo pravidelne na pozadí celého systému, je potrebné vytvoriť tabuľku, ktorá bude obsahovať informácie o sledovaných serveroch a službách, ktoré sa na nich sa majú sledovať. Táto tabuľka bude tiež obsahovať údaj o tom, kedy bolo monitorovanie do systému

pridané a informácie potrebné k sledovaniu (napr. prihlasovacie meno a heslo na pripojenie k databáze, názov databázy a tabuľku, na ktorú sa chceme pripájať pri sledovaní MySQL).

Druhou tabuľkou potom bude tabuľka užívateľa, ktorá je potrebná pre prihlasovanie užívateľov do systému. Tá obsahuje informácie potrebné pre správu užívateľov a užívateľských účtov. Sú to: identifikátor, meno, priezvisko, prihlasovacie meno a heslo užívateľa.

Poslednou tabuľkou je tabuľka s e-mailami, na ktoré sa budú posielať upozornenia pri výpadkoch alebo chybách serverov.



Obrázok 3.3: ER diagram systému

3.2.4 RRD databáza

RRD databázu tvorí súbor .rrd. Vytváranie týchto súborov je navrhnuté tak, aby každé sledovanie v crontab malo 2 vlastné databázy nameraných hodnôt, ktoré sa vytvárajú automaticky po pridaní monitorovania do systému. Prvá databáza ukladá namerané hodnoty každú minútu. Tieto sa potom použijú pri vykresľovaní hodinových, 6-hodinových a 12-hodinových grafov. Druhá ukladá priemernú hodnotu z posledných 60 nameraných hodnôt raz za hodinu.

Táto databáza sa využíva pri vykresľovaní, denných, týždenných a mesačných grafov. Pri vytváraní týchto databáz je tiež potrebné určiť i minimálnu a maximálnu hodnotu dát. Ak sledovanie zo systému odstránime, vymažú sa aj príslušné súbory s databázami daného monitorovania.

4 Implementácia

V tejto kapitole si popíšeme konkrétnu implementáciu jednotlivých častí systému, použité technológie a algoritmy.

4.1 Jadro systému

Jadro systému tvorí skript crontab.sh, ktorý spúšťa všetky skripty slúžiace na monitorovanie odoziev daných služieb. Tento skript je spúšťaný plánovačom úloh- cron. Jeho obsah sa automaticky generuje pri zmene databázy sledovaní. O túto zmenu sa stará skript crontab_edit.php. Priebeh generovania novej tabuľky úloh sa dá zhrnúť do týchto bodov:

- Načítanie údajov z tabuľky servers, ktorá obsahuje informácie o aktuálne bežiacich monitorovaniach
- Zistenie, aká metóda je pri monitorovaní použitá
- Vygenerovanie príkazu, ktorý spustí skript na meranie odozvy danej služby s parametrami z databázy. Napríklad : Ak z databázy zistíme, že chceme monitorovať službu dns, vygenerujeme nasledujúci riadok:
  `$crontab=$crontab."wget -O /dev/null \"http://localhost/dns/dns_timer.php?host=".$zaznam['host']."\"\n";`
  ktorým do reťazca `$crontab` pridáme spustenie skriptu dns_timer.php s parametrom host. Keďže je to php skript, jeho spúšťanie môžeme zariadiť programom wget. Parameter –O /dev/null zabezpečuje, že stiahnuté stánky sa nám nebudú ukladať a zahlcovať systém. Funkcia skriptu dns_timer.php bude vysvetlená neskôr. Podobným spôsobom generujeme spúšťanie skriptov pre všetky záznamy v databáze.
- Prepísanie pôvodného skriptu crontab.sh práve vygenerovaným skriptom crontab.sh

4.1.1 Monitorované protokoly

Systém mal byť navrhnutý tak, aby podporoval širokú škálu služieb a protokolov. Keďže ich množstvo je príliš vysoké, implementoval som niekoľko najznámejších. Rozšírenie pre monitorovanie ďalších služieb a protokolov v tomto systéme je pomerne jednoduché. Na ukážku funkčnosti systému som teda vybral a implementoval monitorovanie týchto protokolov: ICMP, HTTP, FTP, SSH, DNS, MySQL, SMTP, POP3.

4.1.2 ICMP

Keďže bol tento protokol už popísaný v úvode práce ako základný prostriedok, ktorý sa využíva pri monitorovaní sieťovej komunikácie, len si zopakujeme, že slúži na posielanie chybových správ o IP komunikácii. Najznámejším programom, ktorý tento protokol využíva je PING. PING periodicky odosiela prostredníctvom siete packety na konkrétnu IP a čaká na ich návrat. Po ich návrate vypisuje dobu, ako dlho packetu trvalo kým sa vrátil. Ping je teda vhodným prostriedkom na meranie odoziev ICMP protokolu. V systéme sa o volanie programu stará skript icmp_timer.php, ktorý volá program ping s parametrom –c 1. To nám zabezpečí, že sa bude posielať len jeden packet. Nameraná hodnotu potom skript uloží pomocou rrd_update.php do konkrétnych databázy .rrd daného serveru.

4.1.3 HTTP

Ďalším z najznámejších a najpoužívanejších protokolov je HTTP [7]. HTTP alebo HyperText Transfer Protocol je jednoduchý aplikačný protokol, ktorý sa používa predovšetkým k prenosu hypertextových dokumentov a obrázkov. Je postavený na princípe dotaz, odpoveď. Akákoľvek aktivita je vyvolaná zo strany klienta. Server počúva a odpovedá na dotazy štandardne na porte číslo 80. Jednoduchý http dotaz môže teda vyzerať nasledovne:

```
"GET "+ url + " HTTP/1.1\r\nHost: "+ host + "\r\n\n\n"
```

kde `GET` je http metóda, ktorá hovorí serveru, že chce získať nejaký dokument. `url` je potom absolútna alebo relatívna cesta k súboru, ktorý chceme získať. Nasleduje verzia protokolu `HTTP/1.1,` koniec riadku a určuje ktorej domény sa tento dotaz týka. Na jednom fyzickom serveri môže byť totižto viacero domén. Dotaz je ukončený prázdnym riadkom.

Takýto dotaz sa posiela serveru i v našom prípade pomocou programu webclient, ktorý som implementoval v jazyku C++. Program funguje tak, že sa pripojí na server, ktorý dostane ako parameter, na port 80 a pošle mu spomínaný dotaz. Vyčká na odpoveď a ukončí sa. Tento program som implementoval kvôli porovnaniu jeho efektívnosti voči štandardným unixovým programom. I keď som zistil, že implementácia takýchto programov pre všetky protokoly by bola takpovediac „zbytočná“, v systéme som ho ponechal, ako názornú ukážku.

Druhý spôsob ako poslať serveru dotaz GET na http protokol a získať webovú stránku je už niekoľkokrát v príkladoch spomínaný program WGET.

O oba tieto spôsoby sledovania sa stará skript http_timer.php. Ako parametre sú mu predávané server a metóda, ktorú má na monitorovanie použiť. Podobne ako aj pri všetkých

ostatných protokolov a služieb namerané hodnoty skript ukladá do konkrétnych .rrd databáz sledovaného servru pomocou rrd_update.php

4.1.4 FTP

FTP – File Transfer Protocol [8] je aplikačný protokol, ktorý slúži na transport a kopírovanie súborov medzi počítačmi v sieti pričom na týchto počítačoch nemusí byť rovnaký operačný systém. Využíva dva porty: TCP 20 a TCP 21. K prenosu dát slúži port 20 a riadenie tohto prenosu a prenosu FTP príkazov slúži port 21. Klient sa pripája k serveru, kde môže po prihlásení prevádzať rôzne operácie. Na pripojenie k ftp serveru nám v unixových systémoch slúži program ftp, ktorý sa pripojí na daný port na serveri a riadi komunikáciu s ním.

V systéme som sa pomocou jednoduchého skriptu check_ftp.sh, s využitým práve ftp, pripájam na ftp server, pošlem mu prihlasovacie údaje a príkazom nlist vypíšem zoznam súborov a adresárov a aktuálnej zložke na serveri. Čas kým sa tieto operácie vykonajú opäť ukladám do .rrd databázy daného serveru. O celý tento priebeh sa stará skript ftp_timer.php, ktorý je s parametrami server, prihlasovacie meno a heslo automaticky spúšťaný z crontabu.

4.1.5 SSH

SSH [9] umožňuje bezpečnú komunikáciu medzi dvoma počítačmi v sieti. Štandardne sa využíva port č. 22. Protokol tiež zabezpečuje autentizáciu oboch účastníkov komunikácie. Na rozdiel od FTP, ktorý heslá posiela v nezabezpečenej forme, ssh komunikácia je šifrovaná. Podobne ako pri protokole ftp aj protokol ssh má v unixových systémoch rovnomenný program ktorý tento protokol využíva.

Skriptom check_ssh.exp sa na vzdialený server pomocou ssh pripojíme, pošleme prihlasovacie údaje a odhlásime. Opäť meriame čas, ako dlho sa tento skript vykonáva a ssh_timer.php, ktorý sa o to stará výsledky potom ukladá do .rrd databáz daného serveru.

4.1.6 DNS

DNS- Domain Name System [10] patrí k protokolom s ktorými sa stretávame každý deň. Jeho hlavnou úlohou sú vzájomné prevody doménových mien a IP adries. Tento protokol na rozdiel od väčšiny protokolov využíva transportné protokoly TCP i UDP . Komunikácia prebieha na porte 53. Ako väčšina ostatných protokolov i komunikácia DNS prebieha formou dotaz odpoveď.

Najčastejšie používaný program na kontrolu DNS je program nslookup. Pomocou nslookup posielame DNS dotazy DNS serveru a kontrolujeme či DNS server odpovedá správne. Môžeme tiež

pomocou neho vystupovať ako iný DNS server a testovať tak i komunikáciu medzi dvoma DNS servermi. Nám však bude stačiť jednoduchý DNS dotaz. pomocou dns_timer.php voláme program `nslookup www.google.sk $host` – týmto hovoríme serveru $host, že chceme aby nám poslal IP adresu pre doménu www.google.sk. Opäť meriame čas, kým server odpovie a namerané hodnoty ukladáme do príslušných .rrd databáz.

4.1.7 MySQL

MySQL som si vybral preto, že je to asi najznámejší a najpoužívanejší open source databázový systém a výpadky SQL serveru môžu mať v praxi obrovské následky. Komunikácia prebieha na porte 3306 pomocou jazyka SQL.

O meranie časových odoziev sql serveru sa v systéme stará skript mysql_timer.php. Využíva php funkcie: mysql_connect, mysql_select a mysql_query. Komunikácia, ktorej čas meriame zahŕňa pripojenie sa k databáze, výber tabuľky a select nad tabuľkou. Nameraná čas opäť ukladáme do príslušných .rrd databáz monitorovaného serveru.

4.1.8 SMTP

SMTP alebo tiež Simple Mail Transfer Protocol [11] je protokol určený na doručovanie správ elektronickej pošty pomocou priameho spojenia medzi odosielateľom a adresátom. Poštový server beží obvykle ako démon a naslúcha na porte č. 25. K tomuto portu sa pripájajú klienti a komunikáciou vo forme dotaz odpoveď predávajú správy k doručeniu.

V mojej implementácii používa skript smtp_timer.php, ktorý sa stará o meranie odoziev smtp služby a ich ukladanie do .rrd databázy, program telnet na pripojenie sa k portu 25 na danom serveri. Potom prebieha základná komunikácia, tj. SMTP dotaz HELO, ktorým sa klient na serveri autentizuje a výpis informácií o smpt príkazom HELP.

4.1.9 POP3

Ďalším, často využívaným protokolom, ktorý pracuje s elektronickou poštou je Post Office Protocol – POP3 [12]. Slúži na sťahovanie e-mailových správ zo vzdialeného serveru na klienta. Komunikácia opäť prebieha formou dotaz odpoveď na porte 110.

Obdobne ako pri smtp aj pop3_timer.php sa pripája na daný port a simuluje základnú komunikáciu zo serverom pomocou programu telnet . Keďže pop3 vyžaduje prihlásenie, skript check_pop3.exp posiela najskôr serveru prihlasovacie údaje a po úspešnom prihlásení príkazom list

vypíše zoznam mailových správ prihláseného užívateľa na serveri. Namerané časy tejto komunikácie ukladá skript rrd_update.php do príslušných .rrd databáz.

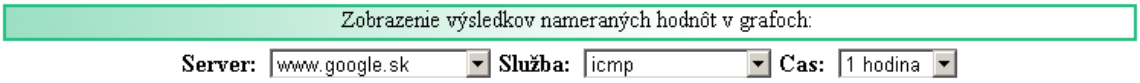
4.2 Webové rozhranie

V tejto kapitole si popíšeme správu monitorovaných serverov a služieb a zobrazenie nameraných hodnôt pomocou webového rozhrania systému. Na vstup do systému je potrebné prihlásenie. Pre potreby projektu som v databázy vytrovil užívateľa „admin1“ a prístupovým heslom: „heslo_admin“. Na účet bez administrátorských práv poslúži napr. „guest“ s prístupovým heslom „heslo“.

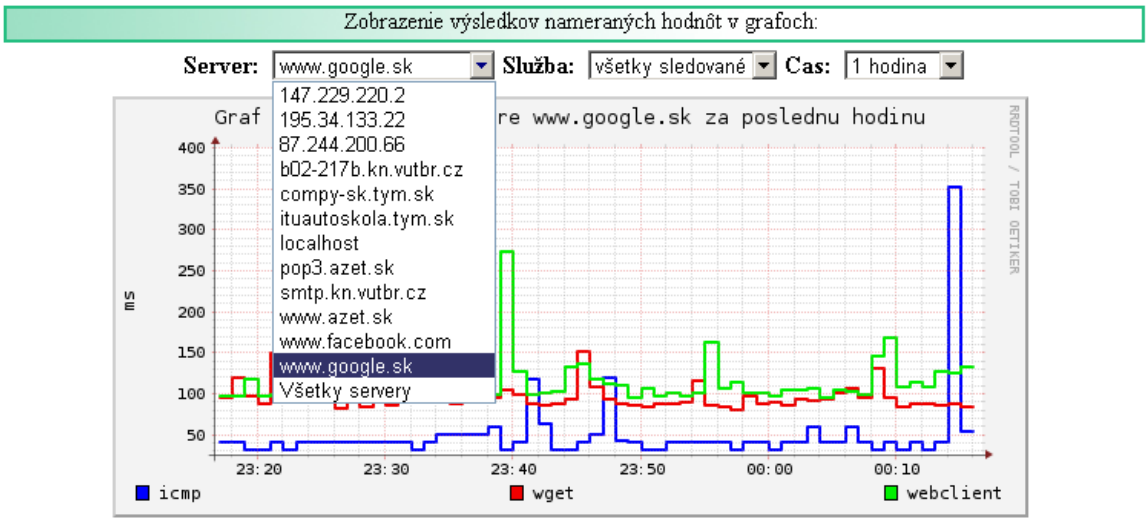
4.2.1 Zobrazovanie výsledkov

Po prihlásení do systému sa zobrazí oddiel Zobrazenie výsledkov, v ktorom sa nachádzajú grafy odoziev na sledovaných serveroch. V hornej časti stránky je už spomínané menu.

Pod hlavným menu sa nachádzajú podrobnejšie možnosti výberu zobrazovaných grafov. Sú zobrazené na obrázkoch 4.1 a 4.2. Podľa potreby si môžeme vybrať iba grafy konkrétneho serveru, konkrétnej služby alebo vyberať z možností zobrazenia 6 druhov grafov. Sú to grafy za poslednú hodinu, za posledných 6 hodín, za posledných 12 hodín, za posledných 24 hodín, grafy za posledný týždeň a grafy posledného mesiaca.

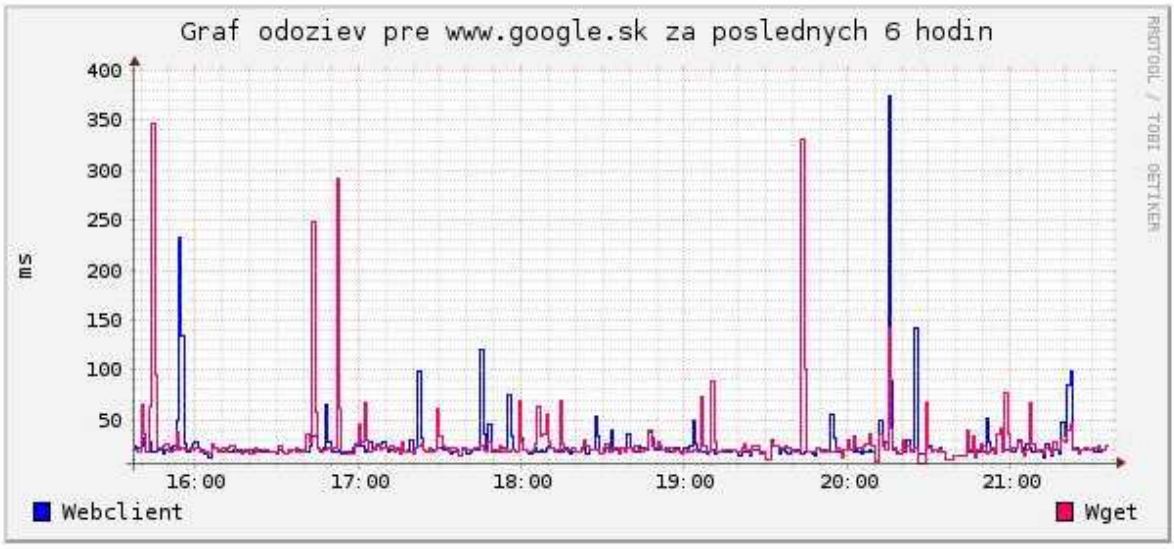


Obrázok 4.1: Formulár výberu zobrazenia grafov



Obrázok 4.2: Príklad možností výberu zobrazenia grafov

Grafy sa (pre lepšie sledovanie aktuálnej situácie na serveri) na otvorenej stránke automaticky obnovujú každé 2 minúty. Na obrázku 4.3 je ukážka výsledného grafu pri meraní http odoziev oboma metódami na serveri www.google.sk.

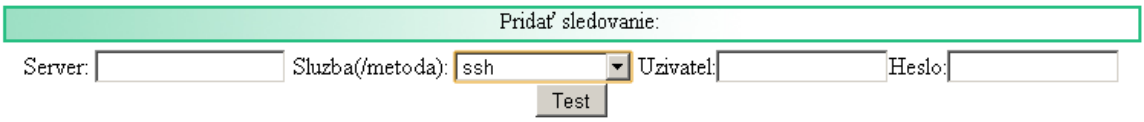


Obrázok 4.3: Príklad výsledného grafu vykresleného systémom

4.2.2 Správa monitorovaní

Veľmi dôležitou časťou celého systému je spravovanie monitorovaní. O túto časť sa na webovom rozhraní stará oddiel, ktorý je v menu nazvaný sledovania.

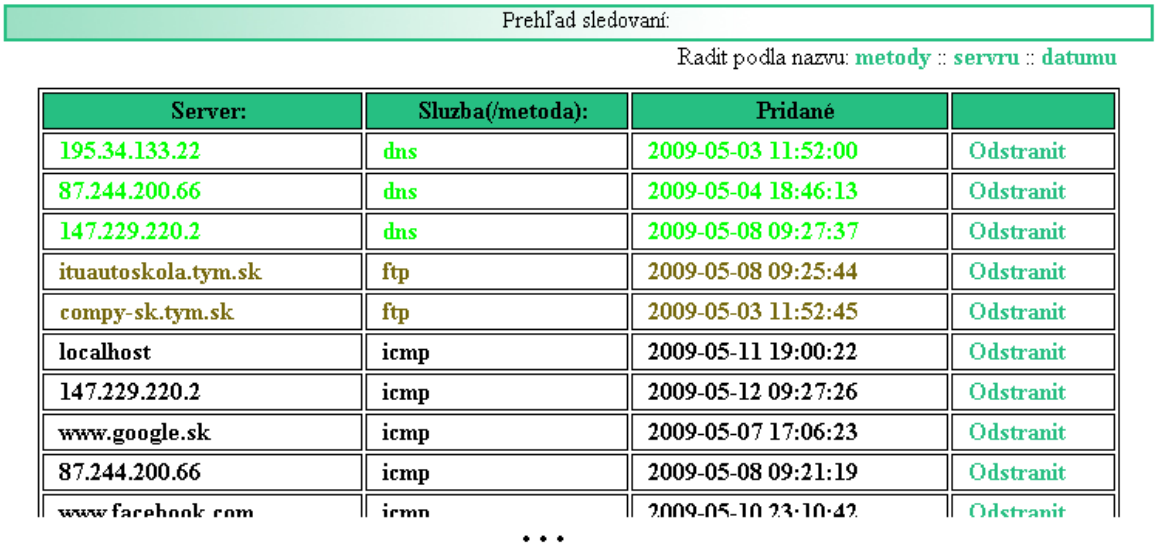
V hornej časti tejto stránky sa nachádza interaktívny formulár pre pridávanie nových monitorovaní do systému (Obrázok 4.4). Po vyplnení formulára je potrebné kliknúť na tlačidlo „TEST“, ktoré overí vyplnené údaje a skontroluje či je daná služba na serveri dostupná a či sa teda dá monitorovať. V prípade že test prebehne úspešne zobrazí sa na stránke nové tlačidlo „PRIDAŤ“, ktorý toto monitorovanie uložíme do databázy.



Obrázok 4.4: Formulár na pridávanie monitorovania

Pod formulárom sa nachádza tabuľka s prehľadom aktuálne bežiacich monitorovaní (Obrázok 4.5), ktorý obsahuje názov alebo IP adresu sledovaných serverov, službu (/metódu ktorou sa daná služba monitoruje) ktorá sa na serveri monitoruje, dátum pridania monitorovania do systému a možnosť odstránenie monitorovania zo databázy. Túto možnosť treba však používať veľmi

opatrne, pretože pri odstránení monitorovania sa automaticky vymažú príslušné rrd databázy a tým stratíme i namerané hodnoty.


Prehľad sledovaní:

Radit podla nazvu: metody :: servru :: datumu

Server:	Sluzba(/metoda):	Pridané	
195.34.133.22	dns	2009-05-03 11:52:00	Odstranit
87.244.200.66	dns	2009-05-04 18:46:13	Odstranit
147.229.220.2	dns	2009-05-08 09:27:37	Odstranit
ituautoskola.tym.sk	ftp	2009-05-08 09:25:44	Odstranit
compy-sk.tym.sk	ftp	2009-05-03 11:52:45	Odstranit
localhost	icmp	2009-05-11 19:00:22	Odstranit
147.229.220.2	icmp	2009-05-12 09:27:26	Odstranit
www.google.sk	icmp	2009-05-07 17:06:23	Odstranit
87.244.200.66	icmp	2009-05-08 09:21:19	Odstranit
www.facebook.com	icmp	2009-05-10 23:10:42	Odstranit

• • •

Obrázok 4.5: Oddiel „Prehľad monitorovaní“

O funkčnosť tohto formulára sa stará viacero skriptov. Po kliknutí na tlačidlo test sa zavolá funkcia: test() z javascriptového script.js. Táto funkcia načíta zadané informácie vo formulári a pošle ich ďalej na spracovanie skriptu test_services.php. Tento následne zistí o akú službu ide, preverí či boli vyplnené všetky údaje potrebné na kontrolu danej služby a podľa služby potom zavolá konkrétny skript alebo program, ktorý overuje funkčnosť služby na serveri.

Konkrétne služby sa potom testujú príslušnými programami, alebo skriptami. Podľa toho či test prebehne úspešne alebo sa vyskytne chyba potom vracia tento skript chybovú správu, alebo oznámenie o úspešnom overení s tlačidlom „PRIDAŤ“ funkcii test, ktorá toto oznámenie zobrazí do príslušnej časti webovej stránky.

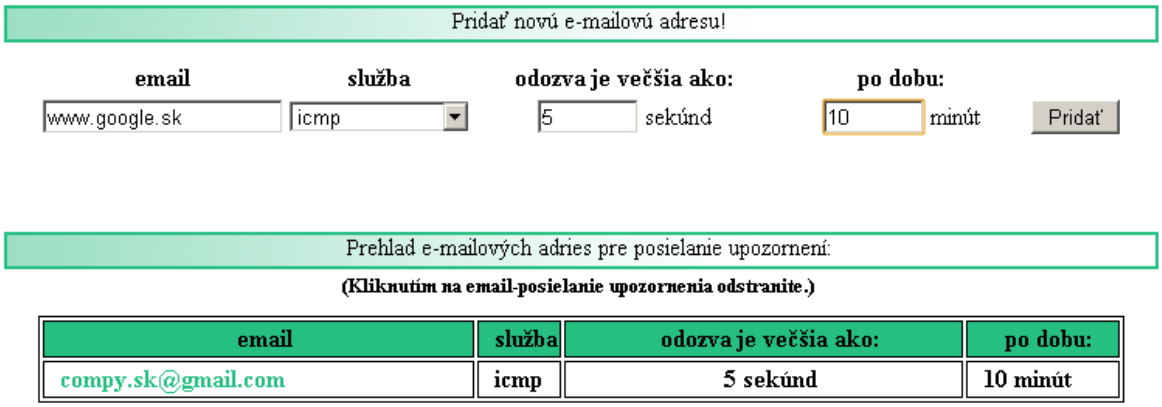
Pridanie monitorovania prebieha tak, že sa najskôr informácie o monitorovaní uložia do SQL databázy a následne sa spustí skript crontab_edit.php, ktorý bol už spomínaný v popise funkčnosti jadra systému.

4.2.3 Správa upozornení

Ďalší oddiel systému tvorí správa upozornení (Obrázok 4.6). V tejto časti majú užívatelia možnosť prezerať si zoznam e-mailov, na ktoré sa posielajú upozornenia v prípade výpadku niektorého zo serverov. Administrátori majú tiež možnosť e-mailovú adresu do zoznamu pridať, alebo odstrániť. Pri pridávaní novej mailovej adresy musí užívateľ nastaviť službu, o ktorej chce dostávať informácie v prípade nedostupnosti, hranicu odozvy, ktorá sa bude považovať za odozvu

nedostupného serveru a čas ako dlho musí server tieto odozvy dosahovať, aby sa odoslal e-mail. Pred samotným pridaním emailu, do systému javasctiprová funkcia `check_mail()` skontroluje správnosť zadaného mailu a potom informácie z formulára predá skriptu add_mail.php , ktorý ich uloží do databázy.

O posielanie mailov sa stará skript send_mail.php, ktorý je pridaný na koniec každého zo skriptov , ktoré merajú časové odozvy serverov. Tento skript si z SQL databázy načíta informácie aké odozvy, ktorej služby, za aký čas a na ktorý e-mail sa má upozornenie poslať. Potom z .rrd databázy načíta posledných n nameraných hodnôt, kde n je čas nedostupnosti serveru. Tieto hodnoty porovná s hranicou pri ktorej sa má odoslať mail s upozornením a ak sú všetky tieto hodnoty väčšie mail sa odošle. Odosielanie prebieha php funkciou mail().

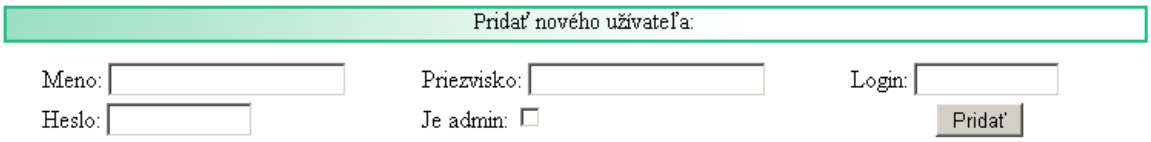


Obrázok 4.6: Oddiel „Správa upozornení“

4.2.4 Správa užívateľov

Posledná položka, ktorú v menu systému nájdeme je Správa užívateľov. Nachádza sa tu tabuľka s prehľadom zaregistrovaných užívateľov s možnosťou ich odstránenia a tiež formulár na registráciu nových užívateľov. Túto možnosť majú však sprístupnenú len užívatelia s administrátorskými právami.

Formulár pre pridanie nového užívateľa (Obrázok 4.7) obsahuje: meno, priezvisko, prihlasovacie meno (login), heslo a zaškrtávacie políčko „Je Admin“. Pri zaškrtnutí tohto políčka dostáva registrovaný užívateľ v systéme administrátorské práva. Po kliknutí na tlačidlo pridať javascriptová funkcia overí, či boli vyplnené všetky požadované políčka a či má prihlásený užívateľ administrátorské práva na pridanie nového užívateľa. V prípade že nie je jedna z týchto podmienok splnená, užívateľ je upozornený a akcia sa nevykoná. V opačnom prípade sa skript add_user.php postará u uloženie informácií do SQL databázy.



Obrázok 4.7: Formulár na pridanie nového užívateľa

5 Záver

Cieľom tejto práce bolo oboznámiť sa so základnými princípmi využívanými pri monitorovaní sietí a sieťových aplikácií a na základe zistených informácií navrhnúť systém ktorý takéto monitorovanie prevádza. Tento návrh sa podarilo úspešne implementovať a otestovať na náhodne vybraných serveroch.

Pri monitorovaní sietí sa využívajú dva prístupy. Prvým je aktívne monitorovanie, ktoré spočíva v simulovaní užívateľských akcií. Jeho najväčšou výhodou je, že informácia o probléme je známe takmer okamžite po tom ako nastal. Tento prístup som využil pri mojej implementácii. Druhý prístup, ktorý sa nazýva aj pasívnym monitorovaním je založený na monitorovaní skutočných užívateľských aktivít na serveroch. Najlepšími monitorovacími nástrojmi v súčasnosti sú tie, ktoré využívajú kombináciu oboch týchto prístupov.

Simulácia užívateľských aktivít so servermi je v systéme implementovaná pomocou štandardných unixových programov ako napr. ping, wget, nslookup, telnet. Pomocou nich sa monitorovacie skripty pripájajú k danej službe a namerané hodnoty ukladajú do rrdtool databáz. Využitie rrdtool na ukladanie sa ukázalo ako najvhodnejšie riešenie, ktoré poskytlo prehľadné zobrazovanie nameraných výsledkov v grafoch.

Správu monitorovaní zabezpečuje jednoduché webové rozhranie s intuitívnym ovládaním. Vystupujú tu dva typy užívateľov: administrátori a hostia. Administrátori majú pomocou webového rozhrania možnosť pridávať a odoberať monitorované serveri a služby na nich. Môžu si tu tiež nastaviť upozorňovanie e-mailami v prípade výpadku konkrétnej služby na niektorom zo sledovaných serverov. Kompletnosť webového rozhrania dopĺňa správa užívateľov s možnosťami pridávať nových užívateľov, nastavovať im administrátorské práva a tiež odstraňovať užívateľov zo systému. Užívatelia bez administrátorských práv majú možnosti obmedzené iba na prezeranie všetkých týchto informácií.

Systém vykresľuje z nameraných hodnôt jednoduché grafy, v ktorých je iba informácia aký graf z ktorej služby sa zobrazuje. Keďže rrdtool je mohutným nástrojom na uchovávanie rôznych hodnôt a ich zobrazovanie v grafoch , ktorý poskytuje veľa možností a nastavení, ponúkajú sa v tejto oblasti mnohé rozšírenia. Mohli by to byť napríklad zobrazovanie maximálnej, minimálnej a priemernej nameranej odozvy daného serveru, alebo interaktívne vytváranie kombinovaných grafov viacerých služieb i viacerých serverov. Rozhranie síce poskytuje možnosť zobrazenia všetkých monitorovaných služieb na jednom serveri do jedného grafu, ale tento prístup by poskytol

napríklad možnosť porovnania si odoziev rovnakej služby na viacerých serveroch a celkovo väčšiu flexibilitu pri zobrazovaní grafov.

Nepochybne vhodným rozšírením by bola i implementácia monitorovacích skriptov pre sledovanie ďalších sieťových aplikácií a služieb, ktorých monitorovanie môže mať v praxi veľký význam. Optimálnym riešením by bola možnosť pridávania týchto skriptov do systému pomocou webového rozhrania. Užívateľ, ktorý by chcel monitorovať službu, ktorej monitorovanie zatiaľ nie je v systéme implementované by pomocou webového rozhrania nahral do systému skript, ktorým by sa daná služba monitorovala, a hneď by mohol pridávať i monitorovania tejto služby a zobrazovať si namerané výsledky v grafoch.

Literatura

[1] Sobell, Mark G.: *Mistrovstvív Linuxu Príkazový řádek, shell, programování.* Brno: Computer Press, a.s., 2007, ISBN 978-80-251-1726-2

[2] *RRDtool Documentation.* 2009. [online] Dostupné z URL: http://oss.oetiker.ch/rrdtool/doc/index.en.html (marec 2009)

[3] Skalar, D.: *PHP 5 moduly, rozšíření, akcelerátory*, Brno :Zoner Press,2005, ISBN 80-86815-19-6

[4] Welling, L.: *PHP a MySQL rozvoj webových aplikací*. Praha: Softpress. 2002. ISBN 80-86497-20-8

[5] Postel, J.: *Internet Control Message PROTOKOL.* RFC 792, 1981. [online] Dostupné z URL: http://www.ietf.org/rfc/rfc0792.txt (apríl 2009).

[6] Case, J., Fedor, M., Schoffstall, M., Davin, J.: *A Simple Network Management Protocol.* RFC 1157, 1990. [online] Dostupné z URL: http://www.ietf.org/rfc/rfc1157.txt (apríl 2009).

[7] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Berners-Lee, T.: *Hypertext Transfer Protocol -- HTTP/1.1*. RFC 2068, 1997. [online] Dostupné z URL: http://www.ietf.org/rfc/rfc2068.txt (máj 2009)

[8] Postel, J., Reynolds, J.: *File Transfer Protocol*. RFC 959, 1985. [online] Dostupné z URL: http://www.ietf.org/rfc/rfc0959.txt (máj 2009)

[9] Ylonen, T., Lonvick, Ed. C.: *The Secure Shell (SSH) Authentication Protocol*. RFC 4252, 2006. [online] Dostupné z URL: http://www.ietf.org/rfc/rfc4252.txt (máj 2009)

[10] Mockapetris, P.: *Domain Names – Implementation and specification.* RFC 1035, 1987. [online] Dostupné z URL: http://www.ietf.org/rfc/rfc1035.txt (máj 2009)

[11] Postel, J.: *Simple Mail Transfer Protocol*, RFC 821, 1982. [online] Dostupné z URL: http://www.ietf.org/rfc/rfc0821.txt (máj 2009)

[12] Myers, J., Rose, M.: *Post Office Protocol – Version 3*. RFC 1939, 1996. [online] Dostupné z URL: http://www.ietf.org/rfc/rfc1939.txt (máj 2009)

Seznam příloh

Priložené dvd obsahuje:

- Zdrojové súbory – kompletné zdrojové súbory PHP webového rozhrania a tiež zdrojové súbory programu webclient a skriptov na testovanie s popisom instalácie
- Image systému pre virtuálny počítač pomocou VMware Player – tu na nachádza celý systém v spustiteľnej podobe.
- Inštalačný program pre VMware Player.