

Správa zařízení a sběr provozních dat rozsáhlé sítě

Bc. Roman Ševčík

Diplomová práce
2015



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Roman ŠEVČÍK**
Osobní číslo: **A10399**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Forma studia: **kombinovaná**

Téma práce: **Správa zařízení a sběr provozních dat rozsáhlé sítě**
Téma anglicky: **The Management and Monitoring of Wide Area Networks**

Zásady pro vypracování:

1. Prostudujte protokol SNMP pro správu zařízení a sběr provozních dat – jak jeho specifikaci, tak typickou implementaci na straně managera i agenta.
2. Analyzujte implementaci SNMP kolektorů pro distribuovaný monitoring/správu v několika vybraných open source systémech (Cacti, Zabbix, Nagios, OpenNMS, ZENOSS).
3. Provedte testování výkonu SNMP kolektorů pro distribuovaný monitoring a správu v síti, která obsahuje řádově desítky tisíc zařízení.
4. Vyhodnoťte výsledky výkonnostních testů a analyzujte příčiny výsledků ve zdrojovém kódu daného kolektoru.
5. Pokud ani jeden z kolektorů nebude mít dostatečný výkon, proveďte analýzu možností implementace vlastního, výkonnějšího kolektoru.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **Net-SNMP**. [online]. [cit. 2015-02-03]. Dostupné z: <http://www.net-snmp.org>
2. **MAURO, Douglas R a Kevin J SCHMIDT**. **Essential SNMP**. 2nd ed. Beijing: O'Reilly, 2005, 442 s. ISBN 05-960-0840-6.
3. **BOUŠKA, Petr**. **SNMP – Simple Network Management Protocol**. In: [online]. [cit. 2015-02-03]. Dostupné z: <http://www.samuraj-cz.com/clanek/snmp-simple-network-management-protocol>
4. **BIGELOW, Stephen J**. **Mistrovství v počítačových sítích: správa, konfigurace, diagnostika a řešení problémů**. Vyd. 1. Překlad Petr Matějů. Brno: Computer Press, 2004, 990 s. ISBN 80-251-0178-9
5. **KRETCHMAR, James M**. **Administrace a diagnostika sítí: pomocí OpenSource utilit a nástrojů**. 1. vyd. Brno: Computer Press, 2004, 216 s. ISBN 80-251-0345-5.
6. **OREBAUGH, Angela**. **Wireshark a Ethereal: kompletní průvodce analýzou a diagnostikou sítí**. Vyd. 1. Brno: Computer Press, 2008, 444 s. ISBN 978-80-251-2048-4.
7. **PERKINS, David T**. **RMON : remote monitoring of SNMP-managed LANs**. Upper Saddle River, NJ: Prentice Hall PTR, c1999, xiii, 440 p. ISBN 0-13-096163-9.
8. **STALLINGS, William a William STALLINGS**. **SNMP, SNMPv2, SNMPv3, and RMON 1 and 2**. 3rd ed. Reading, Mass.: Addison-Wesley, c1999, xv, 619 p. ISBN 02-014-8534-6.

Vedoucí diplomové práce:

Ing. Tomáš Dulík, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

6. února 2015

Termín odevzdání diplomové práce:

15. května 2015

Ve Zlíně dne 6. února 2015



doc. Mgr. Milan Adámek, Ph.D.

děkan



L.S.



doc. Mgr. Roman Jašek, Ph.D.

ředitel ústavu

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně

13.4.2015

.....
podpis autora

ABSTRAKT

Tato práce se zabývá správou zařízení a sběrem provozních dat rozsáhlé sítě. V práci je popsán protokol SNMP a využití jeho metod při sběru dat v síti pomocí monitorovacích systémů. Dále je zde popsána verze SNMP protokolu, způsob zápisu MIB databází a také popis testovaných monitorovacích systémů. V druhé části práce je popsána konfigurace monitorovacích systémů a test jejich vykonnosti, který se odvíjí od počtu zpracovaných klientů rozsáhlé sítě v daném časovém úseku.

Klíčová slova: SNMP, OID, MIB, monitoring, Nagios, Cacti

ABSTRACT

This thesis deals with device management and data collecting in large network. It describes the SNMP protocol and use its methods for collecting data in the network using monitoring systems. It also describes versions of the SNMP protocol, how to write MIB database and also test the description of monitoring systems. The second part describes the configuration of monitoring systems and test of treated clients in large network.

Keywords: SNMP, OID, MIB, monitoring, Nagios, Cacti

Děkuji Ing. Tomáši Dulíkovi Ph.D. za pomoc při vedení diplomové práce.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	9
1 MONITOROVACÍ SYSTÉM	11
1.1 PASIVNÍ SYSTÉMY.....	11
1.2 AKTIVNÍ SYSTÉMY.....	11
2 SNMP.....	12
2.1 ZÁKLADNÍ FUNKCE.....	12
2.2 VERZE SNMP.....	12
2.3 MIB DATABÁZE	12
2.4 SNMP OPERACE.....	13
3 OSTATNÍ SOFTWARE	15
3.1 KVM.....	15
3.2 MYSQL	16
II ANALYTICKÁ ČÁST	16
4 CACTI.....	19
5 NAGIOS	20
III PROJEKTOVÁ ČÁST.....	20
6 POROVNÁNÍ MONITOROVACÍCH SYSTÉMŮ	23
6.1 METODIKA TESTOVÁNÍ	23
7 CACTI.....	25
7.1 INSTALACE.....	25
7.2 KONFIGURACE PRO TESTOVÁNÍ	29
7.3 TESTOVÁNÍ	30
7.3.1 Testování s jedním procesem	30
7.3.2 Testování s více procesy	31
7.4 VYHODNOCENÍ	31
8 NAGIOS	33
8.1 INSTALACE.....	33
8.2 KONFIGURACE PRO TESTOVÁNÍ	34
8.3 TESTOVÁNÍ	37
8.3.1 Testování s jedním procesem	37
8.3.2 Testování s více procesy	37

8.4 VYHODNOCENÍ	38
ZÁVĚR	39
SEZNAM POUŽITÉ LITERATURY	40
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	42
SEZNAM OBRÁZKŮ	43
SEZNAM TABULEK	44
SEZNAM PŘÍLOH	45

ÚVOD

Za posledních 30 let prošel svět technickou revolucí. Počítače na počátku stály miliony a používal je pouze úzký okruh lidí z řad vědců. Roku 1992 byla Česká republika (tehdejší ČSSR) slavnostně připojena k internetu s rychlostí pár kilobitů za sekundu. Dnes vlastní téměř každá domácnost alespoň jeden počítač za dostupnou cenu a je připojena do celosvětové sítě Internet.

Moderní počítačové sítě jsou velký celek obsahující stovky až tisíce prvků. Obecně je můžeme rozdělit do tří skupin:

- koncová zařízení - počítač v domácnosti nebo v kanceláři, servery, tiskárny,
- aktivní síťové prvky - rozbočovač (switch), směrovač (router),
- pasivní síťové prvky - optické, metalické a bezdrátové spoje.

Koncová zařízení služby vytvářejí nebo využívají. Aktivní síťové prvky pomocí pasivních síťových prvků vytvářejí spojení dvou koncových prvků za účelem poskytování a využívání služeb. Příkladem může být kancelářská síť se dvěma počítači, tiskárnou a směrovačem, kdy se mezi koncovými zařízeními nachází jen pár síťových prvků. Pokud se na tento model díváme z pohledu celosvětové sítě, kdy se každé z koncových zařízení může nacházet na jiné straně zeměkoule, zjistíme, že počet síťových prvků mezi nimi jsou desítky až stovky. V případě poškození nebo poruchy aktivního či pasivního síťového prvku dojde k přerušení spojení a také využívání služeb. U malé kancelářské sítě lehce ověříme, který prvek je poškozený. Problém nastává u velkých sítí, které obsahují stovky až tisíce prvků. Zde se nabízí otázka, zda se poruše nedá předejít.

Během monitorování sítě a sběru dat můžeme zdroj problému záhy objevit a lokalizovat nebo dokonce poruše předejít. Jako vhodný nástroj jsme zvolili protokol SNMP, který je jako multiplatformní nástroj podporovaný širokou škálou zařízení jak koncových tak síťových a zároveň SNMP protokol podporuje většina dostupných monitorovacích systémů.

Cílem mé práce je popsat již zmíněný protokol SNMP a možné způsoby monitorování sítě, dále analyzovat zadané monitorovací systémy, provést testování a následně vyhodnocení jejich výkon při zpracování zařízení.

I. TEORETICKÁ ČÁST

1 MONITOROVACÍ SYSTÉM

Monitorovacích systémů lze dnes nalézt velké množství[6]. Od jednoduchých systémů založených například na ICMP protokolu po složité komplexní systémy. Liší se především svým použitím v počítačových sítích. Obecně je můžeme rozdělit do dvou skupin [17] na aktivní a pasivní.

1.1 Pasivní systémy

Pasivní systém přijímá informace, které odesílají koncová zařízení a aktivní prvky. Příkladem může být software Wireshark.

1.2 Aktivní systémy

Aktivní systém sám sbírá data v síti, o která má zájem, takže umožňuje získat rozsáhlejší množství informací. Do této kategorie patří všechny testované systémy v této práci.

V protokolu SNMP lze nalézt pasivní i aktivní přístup k monitorování sítě, protože může přijímat trapy a odesílat request metody.

2 SNMP

SNMP je jednoduchý široce rozšířený standardizovaný protokol, který slouží k získávání nebo nastavování hodnot na určitém zařízení [2]. SNMP funguje asynchronně a je založený na modelu klient/server. Pro komunikaci se používá nedůvěryhodný protokol UDP, který však zatěžuje síť méně než TCP protokol. Pro dotazy se standardně používá port 161 na straně agenta a port 162 pro trapy na straně serveru. Podporu SNMP má celá řada zařízení například aktivní síťové prvky, přístupové body nebo osobní počítače a servery.

2.1 Základní funkce

SNMP protokol potřebuje pro komunikaci dvě strany, managera a agenta.

- Manager - (správce) je systém na serveru, který posílá dotazy a přijímá odpovědi od agentů.
- Agent - je démon běžící na zařízení, který odpovídá na dotazy od managera a posílá mu trapy.

2.2 Verze SNMP

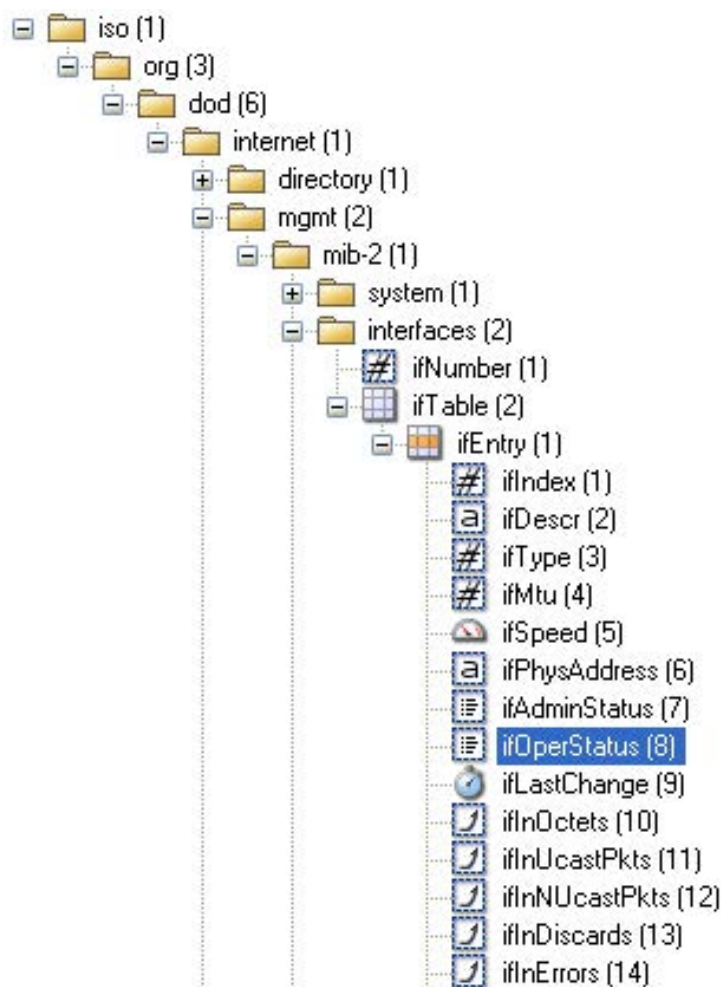
Protokol existuje ve třech verzích 1, 2c a 3 [1].

- Verze 1 - hlavním nedostatkem první implementace protokolu SNMP je autentizace. Používá se community string, což je textové heslo, které není problém na síti odchytnit.
- Verze 2c - tato verze byla vylepšením předešlé a to hlavně v bezpečnosti.
- Verze 3 - aktuální verze protokolu, ve kterém je možné využít šifrování a autentizaci pomocí jména a hesla.

2.3 MIB databáze

Pro označení měřených hodnot se v SMTP protokolu používá OID, který je jednoznačný. Je tvořen posloupností čísel nebo znaků oddělených tečkou. Ty se ukládají v MIB databázi, kterou je myšlen soubor všech objektů dostupných přes SNMP na daném zařízení. Samotnou MIB databázi pro práci s SNMP nepotřebujeme, ale může nám pomoci, pokud nějaké OID hledáme.

OID jsou v MIB databázi řazeny do struktury stromu. Příkladem OID je hodnota statusu síťového rozhraní 1.3.6.1.2.1.2.2.1.8. Na obrázku č. 2.1 je vyobrazena následná struktura stromu.



Obr. 2.1 ISNMP - stromová struktura MIB databáze Zdroj: viz. [9]

Číselná hodnota 1.3.6.1.2.1.2.2.1.8 je reprezentována alfanumerickou hodnotou. iso.org.dod.internet.mgmt.mib-2.interfaces.ifOperStatus [9].

2.4 SNMP Operace

Díky operacím získáváme informace o počítačové síti. Operace udávají způsob přijímání a odesílání informací [1].

- GET - manager požaduje informaci od agenta.
- GETNEXT - manager požaduje od agenta informaci, která je hned za zadaným OID.
- GETBULK - manager požaduje od agenta více informací najednou (například pro všechny síťové rozhraní přepínače).
- SET - manager agentovi nastavuje jednu nebo více informací.

- TRAP - asynchronní oznámení od agenta managerovi. Jedná se o nevyžádanou informaci, například o překročení dané kvóty nebo hodnoty.
- RESPONSE - odpověď od agenta pro managera na GET a SET operaci, která vrací požadovanou hodnotu, popř. číslo chyby.
- INFORM - používá se pro komunikaci mezi managery.
- REPORT - používá se pro indikaci SNMP problému.

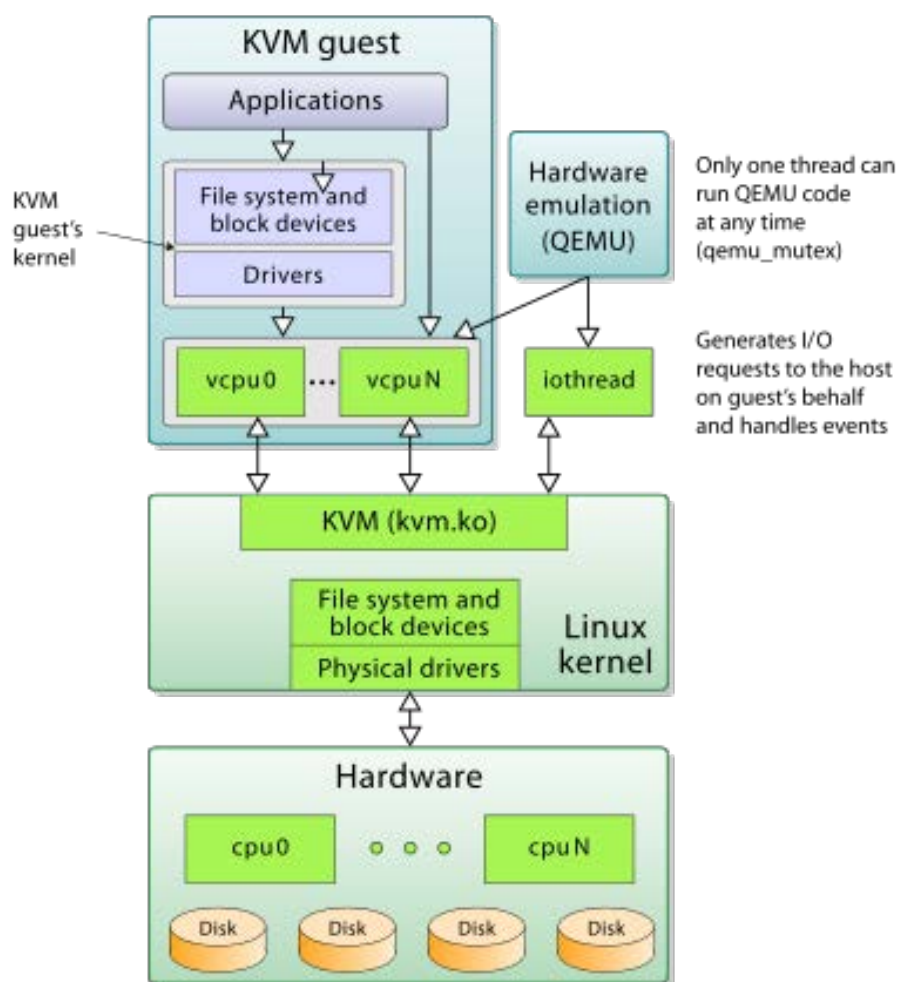
3 OSTATNÍ SOFTWARE

Software, který je v rámci této práce používán.

3.1 KVM

KVM (Kernel-based virtual Machine) je open source software [19] vyvíjený firmou Red Hat a její komunitou. Jedná se o virtualizační řešení na platformě Linux, které se později proměnilo na hypervisor a bylo začleněno do linuxového jádra 2.6.20.

KVM je typ hypervisoru, který umožňuje více operačním systémům sdílet jeden hardware. Hypervisor vytváří virtuální stroje (obr. 3.1) a koordinuje volání pro procesor, paměť, pevný disk, síť a ostatní zdroje připojené skrze hostitelský operační systém.



Obr. 3.1 KVM - přehled úrovní prostředí virtualizace KVM.
Zdroj:Wikipedia

Mezi hostující operační systémy patří BSD, OpenBSD, FreeBSD, Linux, Windows a další. KVM může být nainstalován na všechny procesory typu x86-64 s rozšířenou instrukční sadou pro procesory Intel i AMD.

3.2 MySQL

MySQL je relační databáze typu DBMS, která vychází z deklarativního programovacího jazyka SQL[7] .

MySQL má některá omezení, které jiné systémy nemají (např. Oracle), ale na druhou stranu je rychlý a jednoduchý databázový systém. Práce s ním se dá využít v mnoha programovacích jazycích jako Java, Perl, PHP, C, C++ nebo C#.



Obr. 3.2 Logo databáze MySQL

II. ANALYTICKÁ ČÁST

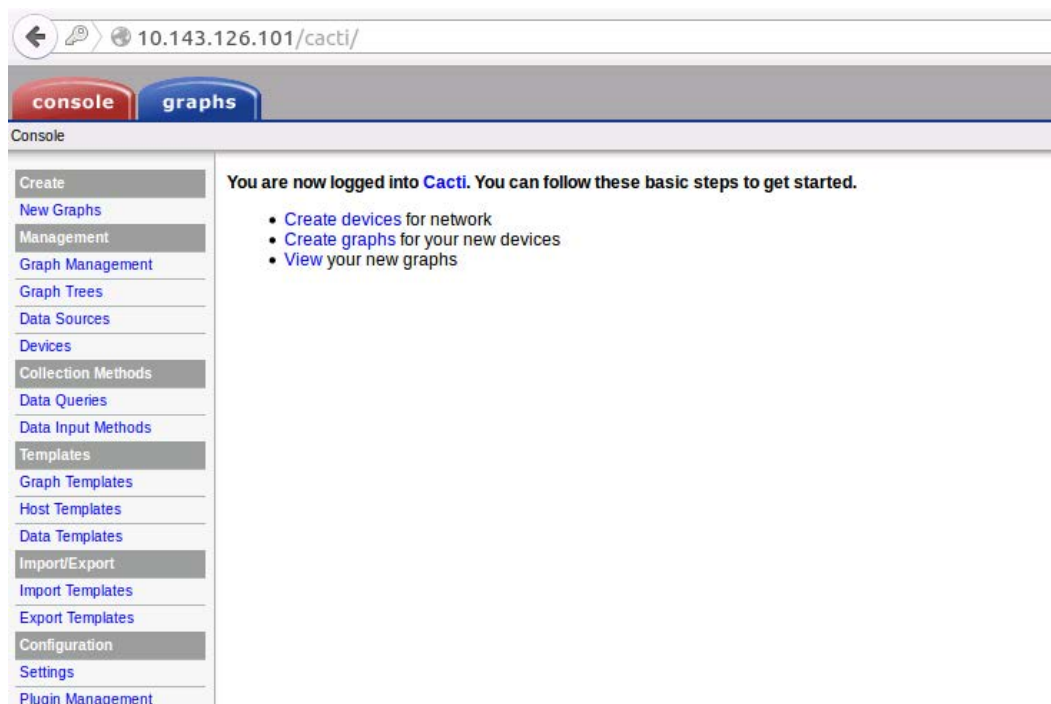
V zadání práce jsou uvedené vybrané open source systémy, které můžeme testovat. Jelikož každý ze systémů je unikátní, tak je třeba pro všechny analyzovat implementaci SNMP kolektoru. Jedná se o systémy Cacti a Nagios. Postupně systémy projdeme a lokalizujeme část systému , která je zodpovědná za sběr dat.

4 CACTI

Cacti [10] je webový nástroj pro monitorování sítě. Umí síť nejen monitorovat, ale i ukládat získaná data do RRD souborů a zobrazovat průběžné grafy. Pomocí Cacti můžeme monitorovat například CPU, paměť, obsazení disků, provoz na síťových kartách. Můžeme také vytvořit vlastní nebo použít předpřipravenou šablonu, jak pro operační systémy, např. Linux nebo Windows, tak pro zařízení jako tiskárny, routery a switche. Cacti můžeme rozdělit na dvě části. Webové rozhraní a skript nebo program na pozadí operačního systému spouštěný v pravidelných intervalech. Jedná se o tzv. poller, který se stará o monitoring systémů v síti a sběr získaných dat a jejich uložení pro grafické zobrazení. Máme na výběr ze dvou pollerů:

- cmd poller - jedná se o implementaci kolektoru v PHP, který je méně výkonný.
- spine poller - kolektor [8] je implantován v jazyce C, je výkonnější než předchozí.

Cacti dovoluje u pollerů nastavit interval spuštění, který je ve výchozím nastavení 5 minut, ale může být i jedna minuta. Dále je možné spustit více procesů najednou a nastavit počet dotazů, který může každý z nich provést. Pokud hostující stroj nemá dostatek výkonu CPU nebo paměti, může se stát, že poller nestihne zkontrolovat všechna zařízení a musíme situaci řešit jejich navýšením nebo změnou intervalu spuštění.



Obr. 4.1 Cacti - úvodní stránka

5 NAGIOS

Existují dva Nagios monitorovací systémy.

- Nagios Core - obsahuje základní knihovny a základ monitorovacího systému.
- Nagios XI - je komerční vydání systému Nagios doplněné o další funkcionality jako zobrazení grafů aj.

Nás bude zajímat open source verze Nagios Core. Nagios [13] obsahuje množství pluginů (může to být program nebo skript v libovolném jazyce), které umí prověřit dostupnost zařízení a jednoduše říct, zda služba funguje nebo ne. Nagios obsahuje podmíněný monitoring a rozlišuje mezi službou běžící na daném stroji a celým strojem. Pokud například zjistí, že daný server není dostupný (neodpovídá na PING dotaz), tak následně vyhodnotí všechny služby jako nedostupné a již je dále netestuje. Umožňuje nastavovat závislosti mezi službami. Pokud například webový server Apache dělá proxy pro jiný webový server a my testujeme dostupnost tohoto jiného serveru, tak v případě že je nedostupný Apache, nemá smysl testovat dostupnost jiného serveru.

Pro svoji monitorovací činnost zpracovává Nagios čtyři druhy testu [14] a to OK/WARNING/ CRITICAL/ UNKNOWN. Plugin, který provádí test vypíše informaci o problému, ale Nagios vyhodnocuje jen návratovou hodnotu OK = 1, WARNING =2, CRITICAL=3, UNKNOWN = 4 a dle toho se rozhodne, jestli se jedná o problém nebo ne. V základní instalaci Nagios funguje jen jako monitoring a není možné ukládat získaná data a vytvářet grafy jako v Cacti. K systému Nagios si ale můžete doinstalovat množství pluginů, které tohle suplují.

Nagios běží jako proces a je dostupný z webového rozhraní (obr. 8.1) pomocí CGI skriptů pro jeho správu. Kolektor startuje samostatný proces pro každého hosta a službu. Pokud například monitorujeme dostupnost HTML a ICPM dostupnost, jedná se o dva procesy. Problém může nastat ve velkých sítích, kdy se vytvoří tisíce procesů a hostující stroj nemá dostatek výkonu nebo paměti. Tento problém můžeme vyřešit navýšením výkonu CPU a paměti a nebo nastavit maximální počet vytvořených spojení [15].

10.143.126.102/nagios3/

Nagios®

General
Home
Documentation

Current Status
Tactical Overview
Map
Hosts
Services
Host Groups
Summary
Grid
Service Groups
Summary
Grid
Problems
Services
(Unhandled)
Hosts (Unhandled)
Network Outages
Quick Search:

Reports
Availability
Trends
Alerts
History
Summary

Nagios® Core™

Nagios® Core™
Version 3.5.1
August 30, 2013
[Check for updates](#)

Get Started

- Start monitoring your infrastructure
- Change the look and feel of Nagios
- Extend Nagios with hundreds of addons
- Get support
- Get training
- Get certified

Quick Links

- [Nagios Library](#) (tutorials and docs)
- [Nagios Labs](#) (development blog)
- [Nagios Exchange](#) (plugins and addons)
- [Nagios Support](#) (tech support)
- [Nagios.com](#) (company)
- [Nagios.org](#) (project)

Obr. 5.1 Nagios - úvodní stránka

III. PROJEKTOVÁ ČÁST

6 POROVNÁNÍ MONITOROVACÍCH SYSTÉMŮ

Při testování výkonu SNMP kolektorů pro monitoring a sběr dat z rozsáhlé sítě se dostáváme do problémů, takovou síť najít. Celosvětová síť sice obsahuje obrovské množství zařízení, ale není možné je jednoduše kontaktovat z mnoha důvodů (např. zařízení nemají povolenou nebo nainstalovanou službu SNMP, jsou zabezpečené nebo nejsou veřejně dostupné). Pokud budeme předpokládat, že kolektory jsou schopny zpracovat tisíce zařízení za minutu nabízí se dva způsoby, jak toho dosáhnout.

- Duplikace zařízení - jelikož není v síti dostatek unikátních zařízení, je potřeba některé duplikovat. Ve skutečnosti se takového zařízení budeme dotazovat vícekrát. Tím můžeme nároky na velikost sítě několikrát snížit. Nevýhodou může být zvýšený provoz v určité části sítě, pokud nerozdělíme duplikovaná zařízení rovnoměrně mezi celou síť. Dále může nastat problém s odpovědí zařízení, pokud se jej bude kolektor dotazovat vícekrát ve stejný čas. Tento problém může nastat hlavně u aktivních síťových prvků jako jsou přepínače a routery, protože mají omezenou paměť.
- Snížení doby měření - další možností je snížení doby měření. Například snížením intervalu z jedné minuty na 10 sekund snížíme požadavek na rozsah sítě 6krát. Zde se ale může objevit vliv aktuálního stavu sítě jako odezva zařízení na SNMP (koncové zařízení mají obecně menší odezvu než aktivní prvky), a proto je vhodné provést více měření.

6.1 Metodika testování

Pro testování výkonu SMTP kolektorů použijeme síť Občanského sdružení Unart ve Slavičíně. V síti se nachází tisíce zařízení, ale nás zajímají zařízení s dostupným SNMP agentem. Takových je v síti kolem 2 tisíc. Jedná se o koncová zařízení (hlavně linuxové servery) a aktivní prvky (přepínače a směrovače infrastruktury, klientské přepínače).

Ve skutečnosti pro nás není tak úplně směrodatné, kolik zařízení zvládneme za minutu zpracovat, ale spíše nás zajímá kolik dat ze zařízení jsme schopni sesbírat. Například z jednoho zařízení nás zajímá vícero informací jako uptime, vytížení CPU, stav obsazenosti disků nebo počet přihlášených uživatelů. V tomto případě se jedná o čtyři SNMP GET dotazy. V SNMP protokolu verze 2 je možné vracet více hodnot najednou (například seznam všech portů na přepínači), takže lze počet dotazů zredukovat. Proto nás při testování bude zajímat počet vrácených hodnot N , které vydělíme průměrným počtem požadovaných hodnot x na zařízení (viz. rovnice 6.1). Výsledkem potom bude průměrný počet zpracovaných zařízení za minutu.

$$y = \frac{N}{\frac{1}{n} \sum_{n=1}^{n=1} x_1} \quad (6.1)$$

7 CACTI

V této části si předvedeme způsob konfigurace a nastavení Cacti pro testování výkonu SNMP kolektoru.

7.1 Instalace

Zde popíšeme postup potřebný k instalaci základního Cacti serveru na Debianu 8 („Jessie“).

Přihlásíme se na server a otevřeme lokální konzoli.

Změníme konzoli na super uživatele.

```
user@cacti~$: su -  
Password:  
root@cacti~$:
```

Ověříme informace o systému.

```
root@cacti:~# uname -a  
Linux cacti 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt9-3~deb8u1 (2015-04-24)  
x86_64 GNU/Linux
```

Aktualizujeme databázi balíčků.

```
root@cacti:~# apt-get update  
Hit http://ftp.cz.debian.org jessie InRelease  
Hit http://ftp.cz.debian.org jessie-updates InRelease  
Hit http://security.debian.org jessie/updates/non-free Translation-en  
Fetched 458 B in 3s (144 B/s)  
Reading package lists... Done
```

Nainstalujeme Cacti pomocí manažeru balíčků.

```
root@cacti:~# apt-get install cacti  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  apache2 apache2-bin apache2-data apache2-utils dbconfig-common file  
  fontconfig fontconfig-config fonts-dejavu-core javascript-common libaio1  
  php5-mysql php5-readline php5-snmp psmisc rename rrdtool sgml-base  
  shared-mime-info snmp ssl-cert xdg-user-dirs xml-core  
0 upgraded, 111 newly installed, 0 to remove and 0 not upgraded.  
Need to get 43.8 MB of archives.  
After this operation, 218 MB of additional disk space will be used.  
Do you want to continue? [Y/n] Y
```

Na defaultní Debian 8 systém nainstalujeme 111 nových balíčků. Potvrdíme jejich stažení a instalaci.

Manažer balíčků se zeptá na několik otázek potřebných k dokončení instalace.

Zadáme MySQL heslo pro superuživatele.

```
While not mandatory, it is highly recommended that you set a password for the MySQL
If this field is left blank, the password will not be changed.
New password for the MySQL "root" user:
abc123[enter]
Repeat password for the MySQL "root" user:
abc123[enter]
```

Vložíme skutečné heslo namísto abc123.

Vybereme webový server, se kterým má být Cacti automaticky nakonfigurováno. V případě, že chceme webový server nakonfigurovat manuálně, vybereme „None“.

```
Web server:
apache [enter]
lighttpd
None
```

Vybereme apache a zmáčkeme enter.

Nakonfigurujeme Cacti s dbconfig-common.

```
The cacti package must have a database installed and configured before
it can be used. This can be optionally handled with dbconfig-common.
Configure database for cacti with dbconfig-common?
Yes[enter]
```

Vybereme Yes a zmáčkeme enter.

Zadáme MySQL heslo super uživatele.

```
Provide the root password for MySQL to create the Cacti database.
Please provide the password for the administrative account with which this
package should create its MySQL database and user.
Password of the database's administrative user:
abc123[enter]
```

Vložíme skutečné heslo namísto abc123.

Zadáme heslo pro nového databázového uživatele Cacti.

```
Please provide a password for cacti to register with the database server.
If left blank, a random password will be generated.
MySQL application password for cacti:
123abc[enter]
Password confirmation:
123abc[enter]
```

Vložíme skutečné heslo namísto abc123.

Instalace je ukončena. Pokračujeme v konfiguraci Cacti. Otevřeme webový prohlížeč a zadáme adresu Cacti serveru ve tvaru **http://<IP>/cacti**.

V našem případě je to **http://10.143.126.101/cacti**.

Po zobrazení stránky (obr. 7.1), vybereme novou instalaci („New Instalation“).

Pokud je to potřeba, v případě že soubor nebyl nalezen („Not Found“), změníme cestu k binárnímu souboru. V naší instalaci (obr. 7.2) dodatečné modifikace nepotřebujeme.



Cacti Installation Guide

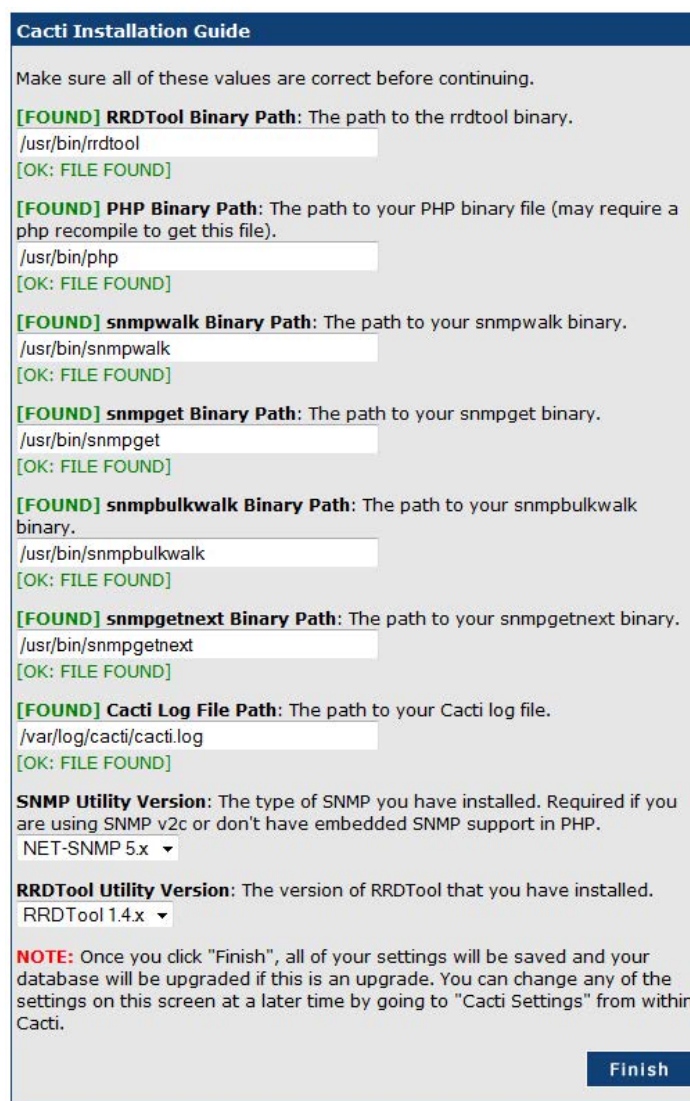
Please select the type of installation

The following information has been determined from Cacti's configuration file. If it is not correct, please edit 'include/config.php' before continuing.

Database User: cacti
Database Hostname: localhost
Database: cacti
Server Operating System Type: unix

Next >>

Obr. 7.1 Instalace Cacti - výběr nové instalace



Cacti Installation Guide

Make sure all of these values are correct before continuing.

[FOUND] RRDTOOL Binary Path: The path to the rrdtool binary.
/usr/bin/rrdtool
[OK: FILE FOUND]

[FOUND] PHP Binary Path: The path to your PHP binary file (may require a php recompile to get this file).
/usr/bin/php
[OK: FILE FOUND]

[FOUND] snmpwalk Binary Path: The path to your snmpwalk binary.
/usr/bin/snmpwalk
[OK: FILE FOUND]

[FOUND] snmpget Binary Path: The path to your snmpget binary.
/usr/bin/snmpget
[OK: FILE FOUND]

[FOUND] snmpbulkwalk Binary Path: The path to your snmpbulkwalk binary.
/usr/bin/snmpbulkwalk
[OK: FILE FOUND]

[FOUND] snmpgetnext Binary Path: The path to your snmpgetnext binary.
/usr/bin/snmpgetnext
[OK: FILE FOUND]

[FOUND] Cacti Log File Path: The path to your Cacti log file.
/var/log/cacti/cacti.log
[OK: FILE FOUND]

SNMP Utility Version: The type of SNMP you have installed. Required if you are using SNMP v2c or don't have embedded SNMP support in PHP.
NET-SNMP 5.x

RRDTOOL Utility Version: The version of RRDTOOL that you have installed.
RRDTOOL 1.4.x

NOTE: Once you click "Finish", all of your settings will be saved and your database will be upgraded if this is an upgrade. You can change any of the settings on this screen at a later time by going to "Cacti Settings" from within Cacti.

Finish

Obr. 7.2 Instalace Cacti - modifikace cest

Přihlásíme se (obr. 7.3) s uživatelem admin a heslem admin.



User Login

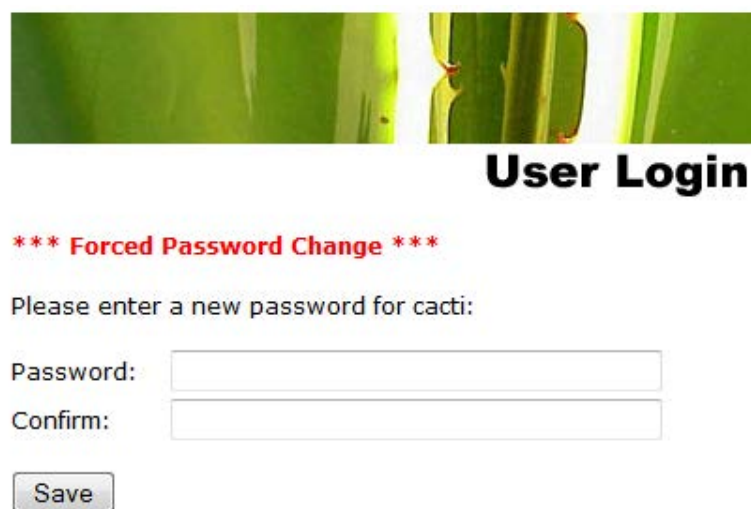
Please enter your Cacti user name and password below:

User Name:

Password:

Obr. 7.3 Instalace Cacti - přihášení

Po přihlášení (obr. 7.4) je nutné změnit heslo.



User Login

***** Forced Password Change *****

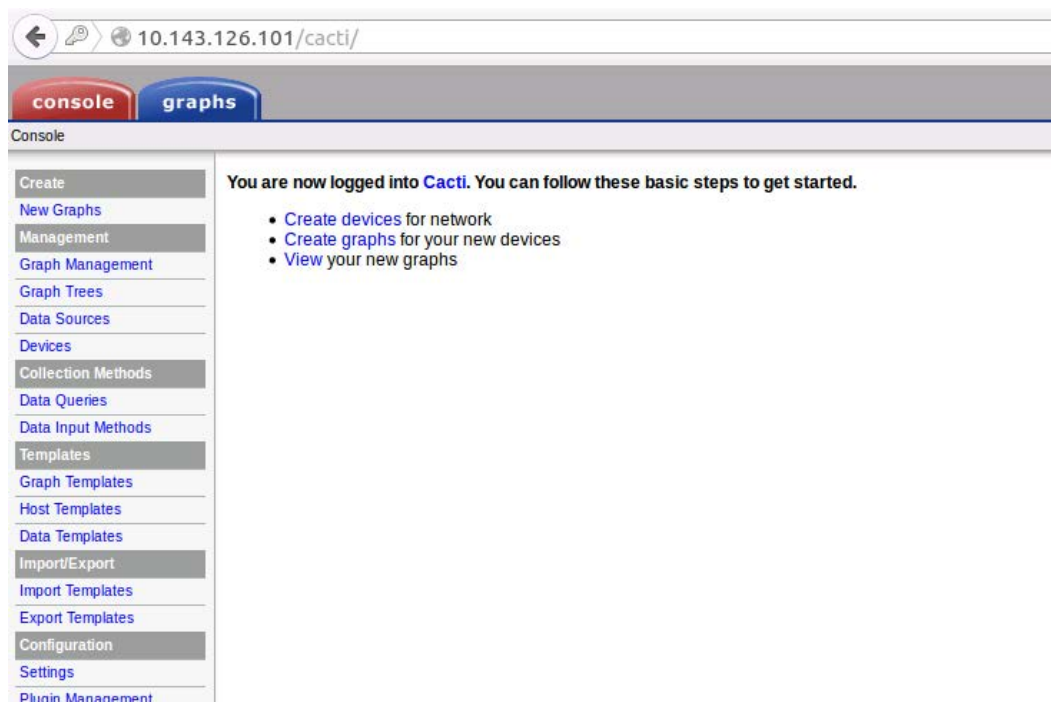
Please enter a new password for cacti:

Password:

Confirm:

Obr. 7.4 Instalace Cacti - změna hesla

Cacti (obr. 7.5) je nyní nainstalováno.



Obr. 7.5 Instalace Cacti - úvod

7.2 Konfigurace pro testování

Pro naše testování musíme nejprve upravit konfiguraci Cacti a primárně naplnit databázi daty. Cacti sice obsahuje přehledné webové rozhraní, ale přidávat a nastavovat přes něj několik tisíc zařízení, je velmi zdoluhavý proces. Naštěstí Cacti obsahuje CLI [16], pomocí kterého data jednoduše importujeme. Nejprve bude potřeba importovat všechna zařízení. Připravíme si soubor, který obsahuje IP adresy všech testovacích zařízení.

```
root@cacti':~# tail -f ip_address
10.143.98.129
10.143.98.131
10.143.98.135
10.143.98.141
10.143.98.2
10.143.99.1
```

Provedeme import dat do Cacti:

```
root@cacti':~# for i in `cat ip_address`; do \
php -q /usr/share/cacti/cli/add_device.php --description=$i --ip=$i \
--template=3 --community=public; \
done
```

Po importu zařízení je třeba vytvořit grafy, pro které se budou sbírat data ze zařízení. Na obsahu dat nám nezáleží. Pro naše potřeby jsme zvolili grafy provozu na všech síťových rozhraní zařízení.

Pro vygenerování grafů zařízení je potřeba získat jejich seznam z databáze. Nás zajímá hlavně jejich ID.

```
root@cacti':~#php -q /usr/share/cacti/cli/add_graphs.php --list-hosts
Known Hosts: (id, hostname, template, description)
2          10.143.0.13      3          10.143.0.13
4          10.143.0.13      3          10.143.0.13
5          10.143.0.14      3          10.143.0.14
6          10.143.0.138     3          10.143.0.138
7..       10.143.0.177     3          10.143.0.177
```

Máme seznam všech ID zařízení a můžeme pro ně vygenerovat grafy.

```
root@cacti':~# for i in `php -q /usr/share/cacti/cli/add_graphs.php \
--list-hosts| awk '{print $1}'`; do \
php -q /usr/share/cacti/cli/add_graphs.php --host-id=$i --graph-type=ds \
--graph-template-id=2 --snmp-query-id=1 --snmp-query-type-id=13 \
--snmp-field=ifOperStatus --snmp-value=Up; \
done
```

Tímto jsme vytvořili veškerou potřebnou infrastrukturu pro testování.

7.3 Testování

Celý test probíhá na virtuálním stroji KVM/QEMU s konfigurací 1x CPU Inte Core 2 Duo 2GHz, 512 MB RAM a Virtio síťová karta 1Gb/s. Jak jsme zmínili v kapitole 4 na straně 19, Cacti má dva možné kolektory neboli pollery. V rámci našeho testování otestujeme cmd poller tj. nativní kolektor napsaný v jazyce PHP.

Pro nativní PHP poller můžeme v nastavení Cacti zvolit jaký maximální počet konkurenčních procesů (obr. 7.6) může poller mít. Pokud je nastavena na „1“, vytvoří se jenom jeden proces a sekvenčně prochází databázi. Pokud je ale vyšší než jedna, je počet zařízení rovnoměrně rozdělen mezi všechny konkurenční procesy.

Cacti Settings (Poller)	
General	
Enabled If you wish to stop the polling process, uncheck this box.	<input checked="" type="checkbox"/> Enabled
Poller Type The poller type to use. This setting will take effect at next polling interval.	cmd.php
Poller Interval The polling interval in use. This setting will effect how often rrd's are checked and updated. NOTE: If you change this value, you must re-populate the poller cache. Failure to do so, may result in lost data.	Every Minute
Cron Interval The cron interval in use. You need to set this setting to the interval that your cron or scheduled task is currently running.	Every Minute
Maximum Concurrent Poller Processes The number of concurrent processes to execute. Using a higher number when using cmd.php will improve performance. Performance improvements in spine are best resolved with the threads parameter	1
Balance Process Load If you choose this option, Cacti will attempt to balance the load of each poller process by equally distributing poller items per process.	<input checked="" type="checkbox"/> Balance Process Load

Obr. 7.6 Cacti

7.3.1 Testování s jedním procesem

První test provedeme pouze s jedním procesem po dobu maximálně jedné minuty. Výsledkem je počet zpracovaných hodnot námi nakonfigurovaných zařízení.

```
SYSTEM STATS: Time:58.1775 Method:cmd.php Processes:1
Threads:N/A Hosts:10142 HostsPerProcess:10142
DataSources:10858 RRDsProcessed:599
```

Celkem jsme získali 599 hodnot během jedné minuty. Pokud dosadíme do rovnice 6.1, získáme průměrný počet zpracovaných zařízení.

$$\frac{599}{6} = \text{cca } 100$$

7.3.2 Testování s více procesy

S pěti procesy

```
SYSTEM STATS: Time:58.2638 Method:cmd.php Processes:5
Threads:N/A Hosts:10142 HostsPerProcess:2029
DataSources:10852 RRDsProcessed:2370
```

$$\frac{2370}{6} = \text{cca } 395$$

S deseti procesy

```
SYSTEM STATS: Time:58.7557 Method:cmd.php Processes:10
Threads:N/A Hosts:10142 HostsPerProcess:1015
DataSources:10852 RRDsProcessed:2782
```

$$\frac{2782}{6} = \text{cca } 464$$

S dvaceti procesy

```
SYSTEM STATS: Time:57.9401 Method:cmd.php Processes:20
Threads:N/A Hosts:10142 HostsPerProcess:508
DataSources:10864 RRDsProcessed:2708
```

$$\frac{2708}{6} = \text{cca } 452$$

7.4 Vyhodnocení

V rámci testování monitorovacího systému Cacti jsme provedli čtyři testy. Jeden test s jedním procesem a tři testy s více konkurenčními procesy. Jak je vidět v tabulce č. 7.1 nejvýkonější byl PHP kolektor při deseti konkurenčních procesech. Bohužel už i při pěti konkurenčních procesech byl server vytížen na téměř 100%. Při dvaceti konkurenčních procesech byl server tak přetížen, že se PHP kolektor zpomalil.

Pokud se podíváme do zdrojového kódu kolektoru, zjistíme, že kolektor docela často čte a zapisuje do databáze. Právě zápis je z hlediska výkonu databáze nejnáročnější a

Tab. 7.1 Cacti - výsledky testů

<i>Počet procesů</i>	1	5	10	20
<i>Prům. počet zařízení</i>	100	395	464	452

zkonsumuje nejvíce systémových zdrojů. Při více jak deseti konkurenčních procesech databáze standardně vytěžovala procesor na více než 65%. Řešením problému nízké efektivity by mohla být eliminace čtení a zápisu do databáze na minimum. Jelikož se popis zařízení a toho, co se má monitorovat příliš často nemění (mění se hlavně data získaná ze zařízení), bylo by možné jednou za delší časový interval vygenerovat konfigurační soubor pro kolektor, který by obsahoval přímo kód na určení zařízení, ze kterého se má hodnota s daným OID získat a do které RRD databáze se má zapsat. V případě více konkurenčních procesů se konfigurace rozdělí do více souborů.

8 NAGIOS

8.1 Instalace

Zde popíšeme instalaci základního Nagios Core systému na Debianu 8 („Jessie“).

Přihlásíme se na server a otevřeme lokální konzoli.

Změníme konzoli na super uživatele.

```
user@nagios~$: su -  
Password:  
root@cacti~$:
```

Ověříme informace o systému.

```
root@nagios:~# uname -a  
Linux cacti 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt9-3~deb8u1 (2015-04-24)  
x86_64 GNU/Linux
```

Aktualizujeme databázi balíčků.

```
root@cacti:~# apt-get update  
Hit http://ftp.cz.debian.org jessie InRelease  
Hit http://ftp.cz.debian.org jessie-updates InRelease  
Hit http://security.debian.org jessie/updates/non-free Translation-en  
Fetched 458 B in 3s (144 B/s)  
Reading package lists... Done
```

Nainstalujeme Nagios pomocí manažeru balíčků.

```
root@nagios:~# apt-get install nagios3 nagios-plugins nagios-nrpe-plugin  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  apache2 apache2-bin apache2-data apache2-utils bind9-host bsd-mailx dbus  
  dnstools exim4-base exim4-config exim4-daemon-light file fontconfig-config  
  rename rpcbind samba-common samba-common-bin samba-libc sgml-base smbclient  
  snmp ssl-cert xml-core  
0 upgraded, 157 newly installed, 0 to remove and 0 not upgraded.  
Need to get 49.8 MB of archives.  
After this operation, 187 MB of additional disk space will be used.  
Do you want to continue? [Y/n] Y
```

Nainstalujeme 157 nových balíčků na defaultní Debian 8 systém, potvrdíme jejich stažení a instalaci.

Zadáme heslo pro webovou administraci systému Nagios.

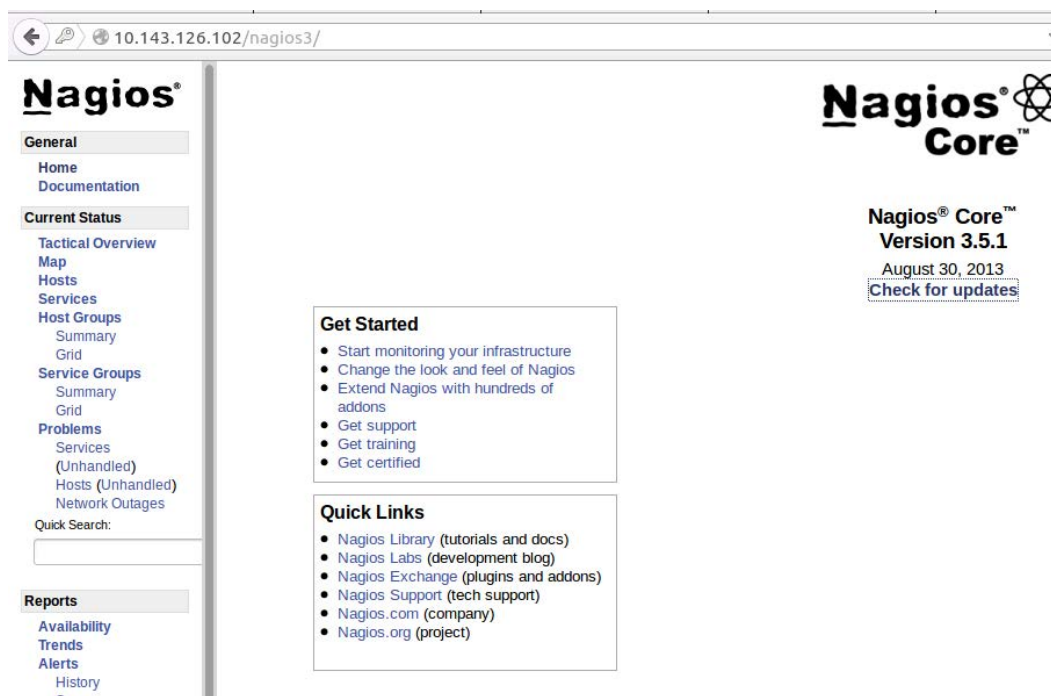
```
Please provide the password to be created with the "nagiosadmin" user.
This is the username and password you will use to log in to your nagios
installation after configuration is complete. If you do not provide
a password, you will have to configure access to nagios yourself.
Nagios web administration password:
abc123[enter]
Repeat Nagios web administration password:
abc123[enter]
```

Vložíme skutečné heslo namísto abc123.

Po dokončení instalace otevřeme webový prohlížeč a zadáme adresu Nagios serveru ve tvaru **http://<IP>/nagios3**.

V našem případě je to **http://10.143.126.102/nagios3**.

Nagios (obr. 8.1) je nyní nainstalován.



Obr. 8.1 Instalace Nagiosu

8.2 Konfigurace pro testování

Pro testování výkonnosti je nejdříve potřeba nakonfigurovat Nagios. Nagios Core neobsahuje žádné webové rozhraní pro úpravu a je třeba vše konfigurovat v souborech. Pro jednoduchost je konfigurace upravena pro výchozí soubory Debianu.

Připravíme si soubor, který obsahuje IP adresy všech testovacích zařízení.

```
root@nagios:~# tail -f ip_address
10.143.98.129
10.143.98.131
10.143.98.135
10.143.98.141
10.143.98.2
10.143.99.1
```

Cílem je pro každé zařízení natavit 6 jakýchkoliv testovaných SNMP zdrojů. V souboru `/etc/nagios3/commands.cfg` nadefinujeme 6 nových příkazů. Pro naše účely volíme například OID pro uptime, neboli jak dlouho je dané zařízení dostupné.

```
define command{
    command_name    snmp_uptime1
    command_line    \${USER1}\$/check_snmp -o .1.3.6.1.2.1.2.2.1.11.1 \
-H \${HOSTADDRESS}\$ \${ARG1}\$
}
define command{
    command_name    snmp_uptime2
    command_line    \${USER1}\$/check_snmp -o .1.3.6.1.2.1.2.2.1.11.1 \
-H \${HOSTADDRESS}\$ \${ARG1}\$
}
define command{
    command_name    snmp_uptime3
    command_line    \${USER1}\$/check_snmp -o .1.3.6.1.2.1.2.2.1.11.1 \
-H \${HOSTADDRESS}\$ \${ARG1}\$
}
define command{
    command_name    snmp_uptime4
    command_line    \${USER1}\$/check_snmp -o .1.3.6.1.2.1.2.2.1.11.1 \
-H \${HOSTADDRESS}\$ \${ARG1}\$
}
define command{
    command_name    snmp_uptime5
    command_line    \${USER1}\$/check_snmp -o .1.3.6.1.2.1.2.2.1.11.1 \
-H \${HOSTADDRESS}\$ \${ARG1}\$
}
```

Dále definujeme služby pro příkazy výše. V souboru `/etc/nagios3/conf.d/services_nagios2.cfg` nadefinujeme 6 nových služeb.

```
define service {
    hostgroup_name test-servers
    service_description Benchmark test 0
    check_command snmp_uptime0
    use generic-service
    notification_interval 0
}
define service {
    hostgroup_name test-servers
    service_description Benchmark test 1
    check_command snmp_uptime1
    use generic-service
    notification_interval 0
}
define service {
    hostgroup_name test-servers
```

```

        service_description Benchmark test 2
        check_command snmp_uptime2
        use generic-service
        notification_interval 0
    }
define service {
    hostgroup_name test-servers
    service_description Benchmark test 3
    check_command snmp_uptime3
    use generic-service
    notification_interval 0
}
define service {
    hostgroup_name test-servers
    service_description Benchmark test 4
    check_command snmp_uptime4
    use generic-service
    notification_interval 0
}
define service {
    hostgroup_name test-servers
    service_description Benchmark test 5
    check_command snmp_uptime5
    use generic-service
    notification_interval 0
}
}

```

Potřebujeme ještě vytvořit skupinu pro všechny zařízení se jménem test-servers. Všechny zařízení, které do ní budou patřit budou automaticky získávat informace od námi definovaných služeb. V souboru `/etc/nagios3/conf.d/hostgroups_nagios2.cfg` definujeme skupinu jako „wild card“, to znamená, že do ní budou automaticky patřit všichni přidání hosté.

```

define hostgroup {
    hostgroup_name test-servers
        alias          Test Servers
    members          *
}

```

Pro generování souboru zařízení nám pomůže předpřipravený skript `import.sh`, který vygeneruje seznam hostů v Nagios formátu.

```

#!/bin/bash
# Generate Nagios host file.
for i in `cat ip_address`;
do
    date=$(date +"%T");
    echo "define host{
    use          generic-host
    host_name    \${date}"-"\"$i
    address     \${i
    }
    ";
done

```

Nakonec provedeme `import` hostů do Nagiosu a restartujeme jej. `/etc/localhost_nagios2.cfg`.

```
root@nagios':~# bash import.sh >> /etc/localhost_nagios2.cfg
```

```
root@nagios:~# /etc/init.d/nagios3 restart
[ ok ] Restarting nagios3 (via systemctl): nagios3.service.
```

Tímto jsme vytvořili veškerou potřebnou infrastrukturu pro testování.

8.3 Testování

Celý test probíhá na virtuálním stroji KVM/QEMU s konfigurací 1x CPU Inte Core 2 Duo 2GHz, 512 MB RAM a Virtio síťová karta 1Gb/s.

Nagios podporuje konkurenční procesy. Nastavení se provádí v souboru `/etc/nagios3/nagios.cfg`.

```
root@nagios:~# grep max_concurrent_checks /etc/nagios3/nagios.cfg
max_concurrent_checks=1
```

Pokud je hodnota nastavena na „1“, vytvoří se jenom jeden proces. Pokud je ale vyšší než jedna, je počet zařízení a služeb rozdělen mezi všechny konkurenční procesy. Je-li je rovna „0“, není počet konkurenčních procesů omezen.

8.3.1 Testování s jedním procesem

První test provedeme pouze s jedním procesem. Výsledkem je počet zpracovaných hodnot námi nakonfigurovaných zařízení za minutu.

```
Services Checked: 75
```

Celkem jsme získali 75 hodnot během jedné minuty. Pokud dosadíme do rovnice 6.1, získáme průměrný počet zpracovaných zařízení.

$$\frac{75}{6} = cca 13$$

8.3.2 Testování s více procesy

S dvaceti procesy

```
Services Checked: 611
```

$$\frac{611}{6} = cca 102$$

S sto procesy

```
Services Checked: 841
```

$$\frac{841}{6} = cca 140$$

S neomezeně procesy

Services Checked: 972

$$\frac{972}{6} = \text{cca } 162$$

8.4 Vyhodnocení

V rámci testování monitorovacího systému Nagios jsme provedli čtyři testy. Jeden test s jedním procesem a tři testy s více konkurenčními procesy. Jak je vidět v tabulce č. 8.1 neuvýkonnější byl Nagios kolektor při neomezeně konkurenčních procesech. Po celou dobu měření byl virtuální server vytížen na 100%. Jak procesor, tak paměť.

Tab. 8.1 Cacti - výsledky testů

<i>Počet procesů</i>	1	20	100	neomezeně
<i>Prům. počet zařízení</i>	75	611	841	972

Jelikož po celou dobu testu byl virtuální stroj vytížen na 100% a kolektor byl omezen jeho výkonem, nemůžeme říci, že jedná o maximální hodnoty. Pro zlepšení testu měření je potřeba použít silnější procesor a více paměti.

ZÁVĚR

V této práci jsme se zabývali správou zařízení a sběrem provozních dat rozsáhlé sítě. Popsali jsme si protokol SNMP a jeho využití a dále analyzovali implementaci SNMP kolektorů z několika vybraných open source systémů. Vybrali jsme systémy Cacti a Nagios. V další části jsme provedli testování výkonu SNMP kolektoru Cacti v síti Občanského družství Unart ve Slavičíně. Samotné vyhodnocení výsledků výkonnostních testů nebylo moc příznivé pro nasazení ve velké síti, protože Cacti kolektor dokázal zpracovat jen zlomek požadovaného počtu zařízení. Nagios naopak narazil na limity použitého virtuálního stroje a jeho výsledky nejsou objektivní. V poslední části jsme analyzovali příčiny špatných výsledků ve zdrojovém kódu a navrhli vlastní úpravy vedoucí k vylepšení výkonnosti.

SEZNAM POUŽITÉ LITERATURY

- [1] MAURO, Douglas R a Kevin J SCHMIDT. *Essential SNMP*. 2nd ed. Beijing: O'Reilly, 2005, 442 s. ISBN 05-960-0840-6.
- [2] BOUŠKA, Petr. SNMP - Simple Network Management Protocol. In: *SAMURAJ-cz.com - počítačové sítě, Cisco, Microsoft, VMware, administrace*: [online]. 2006 [cit. 2015-03-14]. Dostupné z: <http://www.samuraj-cz.com/clanek/snmp-simple-network-management-protocol>
- [3] CASE, Jeffrey D., Mark FEDOR, Martin Lee SCHOFFSTALL a Chuck DAVIN. Simple Network Management Protocol (SNMP). In: *Request for comments published by the Internet Engineering Task Force* [online]. 1989 [cit. 2015-03-14]. ISSN 2070-1721. Dostupné z: <http://www.ietf.org/rfc/rfc1157.txt>
- [4] CASE, Jeffrey D., Keith MCCLOGHRIE, Marshall T. Rose ROSE a Steve WALDBUSSER. Introduction to Community-based SNMPv2. In: *Request for comments published by the Internet Engineering Task Force* [online]. 1996 [cit. 2015-03-14]. ISSN 2070-1721. Dostupné z: <http://www.rfc-editor.org/rfc/rfc1901.txt>
- [5] WIJNEN, Bert, Dave HARRINGTON a Randy PRESUHN. An Architecture for Describing SNMP Management Frameworks. In: *Request for comments published by the Internet Engineering Task Force* [online]. 1999 [cit. 2015-03-14]. ISSN 2070-1721. DOI: Randy Presuhn. Dostupné z: <http://www.rfc-editor.org/rfc/rfc2571.txt>
- [6] Comparison of network monitoring systems. *Wikipedia, the free encyclopedia*: [online]. 2001 [cit. 2015-04-14]. Dostupné z: http://en.wikipedia.org/wiki/Comparison_of_network_monitoring_systems
- [7] What is MySQL?. *MySQL :: Developer Zone* [online]. [cit. 2015-05-02]. Dostupné z: <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>
- [8] Spine Information. *Cacti; - The Complete RRDTool-based Graphing Solution* [online]. [cit. 2015-05-10]. Dostupné z: http://www.cacti.net/spine_info.php
- [9] SNMP Tutorial Part 2: Rounding Out the Basics. In: *Network Management & Monitoring Downloads* [online]. [cit. 2015-05-20]. Dostupné z: <http://www.networkmanagementsoftware.com/snmp-tutorial-part-2-rounding-out-the-basics>
- [10] What is Cacti? *Cacti® - The Complete RRDTool-based Graphing Solution* [online]. [cit. 2015-03-14]. Dostupné z: http://www.cacti.net/what_is_cacti.php

-
- [11] What is Zabbix? *Zabbix :: The Enterprise-Class Open Source Network Monitoring Solution* [online]. [cit. 2015-03-14]. Dostupné z: <http://www.zabbix.com/product.php>
- [12] About OpenNMS. *The OpenNMS Project* [online]. [cit. 2015-03-14]. Dostupné z: <http://www.opennms.org/about/>
- [13] About Nagios. *Nagios - The Industry Standard in IT Infrastructure Monitoring*: [online]. [cit. 2015-03-14]. Dostupné z: <http://www.nagios.org/about>
- [14] Host Checks. *Nagios Core Documentation* [online]. [cit. 2015-05-11]. Dostupné z: http://nagios.sourceforge.net/docs/3_0/hostchecks.html
- [15] Service Check Scheduling. *Nagios plugins* [online]. [cit. 2015-05-10]. Dostupné z: <http://nagios.manubulon.com/traduction/docs14en/checkscheduling.html>
- [16] Command Line Scripts. *The Cacti Manual* [online]. [cit. 2015-05-13]. Dostupné z: <http://www.cacti.net/downloads/docs/html/scripts.html>
- [17] MACÁK, Petr. *Kritéria výběru software pro malé a středně velké společnosti* [online]. 2011 [cit. 2015-04-16]. Dostupné z: http://www.cssi.cz/cssi/system/files/all/si_2011_01_10_Macak.pdf. Vysoká škola ekonomická v Praze.
- [18] KRETCHMAR, James M a Kevin J SCHMIDT. *Administrace a diagnostika sítí: pomocí OpenSource utilit a nástrojů*. 1. vyd. Brno: Computer Press, 2004, 216 s. ISBN 80-251-0345-5.
- [19] Kernel Based Virtual Machine. *KVM* [online]. [cit. 2015-05-01]. Dostupné z: http://www.linux-kvm.org/page/Main_Page

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

SNMP	Simple Network Management Protocol
OID	Object Identifier
MIB	Management information Base
ICMP	Internet Control Message Protocol
KVM	Kernel-based Virtual Machine
QEMU	Quick Emulator
DBMS	Database management System
PHP	Hypertext Preprocessor
RRDtool	round-robin database
CPU	Central processing Unit
CGI	Common Gateway Interface
HTML	HyperText Markup Language
CLI	Command-line interface
SQL	Structured Query Language

SEZNAM OBRÁZKŮ

Obr. 2.1	ISNMP - stromová struktura MIB databáze Zdroj: viz. [9]	13
Obr. 3.1	KVM - přehled úrovní prostředí virtualizace KVM. Zdroj:Wikipedia	15
Obr. 3.2	Logo databáze MySQL	16
Obr. 4.1	Cacti - úvodní stránka	19
Obr. 5.1	Nagios - úvodní stránka	21
Obr. 7.1	Instalace Cacti - výběr nové instalace	27
Obr. 7.2	Instalace Cacti - modifikace cest	27
Obr. 7.3	Instalace Cacti - přihášení	28
Obr. 7.4	Instalace Cacti - změna hesla	28
Obr. 7.5	Instalace Cacti - úvod	29
Obr. 7.6	Cacti	30
Obr. 8.1	Instalace Nagiosu	34

SEZNAM TABULEK

Tab. 7.1	Cacti - výsledky testů	32
Tab. 8.1	Cacti - výsledky testů	38

SEZNAM PŘÍLOH

P I. Obsah CD ROM

PŘÍLOHA P I. OBSAH CD ROM

Přiložený CD ROM obsahuje zdrojový kód práce v jazyce \LaTeX .