Výběr a nasazení monitorovacího systému pro účely sledování stavu Virtuální laboratoře počítačových sítí.

Martin Milata
Katedra informatiky FEI, Vysoká Škola Báňská - Technická Univerzita Ostrava
17. listopadu 15
Ostrava-Poruba, Česká republika
martin.milata@vsb.cz

ABSTRACT

S rozvojem a stále intenzivnějším využíváním Virtuální laboratoře počítačových sítí (Virtlab), jejímž cílem je vzdálené zpřístupnění laboratorních prvků pomocí Internetu, významně vzrůstají nároky na zajištění pokud možno nepřetržitého provozu. Chování Virtlabu jako celku, ale i jeho jednotlivých součástí, je proto potřeba neustále sledovat, stav vyhodnocovat a vhodně reagovat na případné abnormality. Příspěvek si klade za cíl seznámit čtenáře s problematikou výběru vhodného monitorovacího systému a jeho nasazením v podmínkách dnes plně distribuovaného systému Virtuální laboratoře počítačových sítí.

Keywords

Virtuální laboratoř počítačových sítí, Virtlab, Nagios, dohledové systémy

1. ÚVOD

Od samotného počátku bylo smyslem projektu Virtuální laboratoře počítačových sítí (Virtlab) vzdálené zpřístupnění mnohdy nákladného laboratorního vybavení pro plnohodnotnou práci nejen studentů z pohodlí Internetového prohlížeče[**?**]. Od zahájení projektu v roce 2005 celý systém prošel dramatickým vývojem, který vedl až k jeho dnešní plně distribuované architektuře[**?**]. Po dvou letech od startu projektu byl také Virtlab ve své pilotní konfiguraci nasazen v rámci spolupráce FEI VŠB-Technické univerzity Ostrava a Obchodně-podnikatelské fakulty Slezské univerzity v Karviné[**?**]. Dnes je využíván v několika vysokoškolských kurzech jako nástroj k procvičování problematiky praktické konfigurace počítačové sítě. Díky integrovaným rozšířením také významně usnadňuje vyhodnocování práce studentů pedagogy.

Vytíženost a požadavky na provozuschopnost celého systému již během vývoje vedly k potřebě zavádění monitorovacích prvků, které informovaly správce o případných problémech. Jak ale narůstala komplexnost celého systému, zvyšovaly se i nároky na potřeby monitorování jeho součástí. Dlouhodobý výpadek jedné komponenty totiž může ve svém důsledku i v distribuovaném prostředí nepříznivě ovlivnit chování zbytku systému. Dříve nasazené mechanizmy informování pověřených osob o vzniklých abnormalitách se postupně stávaly nedostačující a požadavek na nasazení a provoz nezávislého dohledového systému, který by dokázal kontrolovat jak standardní prostředky serverů tak chování specializovaných komponent a zařízení, nabýval na svém významu. V následujících kapitolách bude postupně představena problematika výběru dohledového software vzhledem k požadavkům, které na něj byly kladeny v plánovaném mírně nestandardním nasazení. Rovněž bude stručně popsáno jeho nasazení do testovacího provozu v rámci virtualizovaného testovacího prostředí a následný přechod k ostrému provozu.

2. POŽADAVKY NA MONITOROVÁNÍ KOMPONENT

Předtím než mohl být zvolen pro zamýšlené účely nejvhodnější kandidát, musela být provedena analýza požadavků, které na něj budou kladeny při nasazení v distribuovaném prostředí Virtuální laboratoře počítačových sítí. Zde totiž, na rozdíl od případů nasazení dohledového systému na standardní "dobře známé" služby, bylo potřeba počítat s nutností monitorování komponent, jenž povětšinou do žádného běžného schématu dohledu nezapadnou. Bylo potřeba zvážit, které části Virtlabu budou monitorovány a jakým způsobem bude dohled zajištěn. Vzniklo tak následující hrubé dělení do čtyř skupin

- dostupnost systému (dostupnost vlastního serveru pomocí počítačové sítě, dostupnost služby Secure Shell)
- stav základních prostředků serveru (zatížení serveru, využití diskového prostoru, stav pevných disků, ...)
- kontrola běhu a stavu softwarových komponent Virtlabu (monitorování komponent vzdáleného přístupu ke konzolám, komponent tunelování provozu mezi lokalitami, komponent mazání konfigurací laboratorního vybavení, webového rozhraní Virtlabu, ...)
- kontrola specializovaného hardwarového vybavení (propojování synchronních sériových linek v realizovaných topologiích).

První dvě skupiny nebyly z hlediska sledování nijak neobvyklé a tak požadavky, které z nich na dohledový systém

plynuly, jeho volbu významně neovlivnily. Monitorování stavu komponent ve třetí a čtvrté skupině již nelze provádět jen pomocí standardních dohledových modulů. Jako příklad uveďme komponentu tunelovacího serveru, jejímž úkolem je tunelování provozu mezi jednotlivými zařízeními Virtlabem realizované síťové topologie a to mezi lokalitami Virtlabu nebo v rámci nich. Samotná kontrola přítomnosti běžícího procesu tunelovacího serveru je jednoduchou záležitostí, kterou zvládnou všechny zvažované dohledové systémy[1]. Pokud ale budeme chtít ověřit jeho správnou funkčnost z hlediska předávání síťového provozu, bez vlastního testovacího modulu se neobejdeme[2]. Další požadavky pak byly odvozeny od potřeby uchovávání detailní historie stavů Virtlabu před výskytem abnormality, vhodné formy zaslání informační zprávy o vzniklé události konkrétní osobě či skupině osob nebo snaha o budoucí jednoduchou konfiguraci napojením na systém jednotného generování konfiguračních souborů.

Postupně tak byly definovány následující požadavky, které byly na dohledový systém kladeny

- monitoring dostupnosti fyzických (případně virtuálních) serverů pomocí počítačové sítě
- monitorování standardních služeb a to především (Secure Shell, SMTP, SNMP, Web, ...)
- monitorování vlastních služeb pomocí
  - vhodné parametrizace standardních dohledových modulů
  - vlastních dohledových modulů
- uchovávání detailní historie stavů komponent i Virtlabu jako celku
- pokročilé možnosti notifikace vybraných uživatelů či skupin a to především pomocí e-mailu nebo IM zpráv.
- snadné napojení na systém automatizovaného generování konfiguračních souborů
- open-source řešení

3. VýBěR DOHLEDOVÉHO SYSTÉMU

V okamžiku, kdy byly známy požadavky na hledaný software, započalo se s hledáním vhodných kandidátů, kteří by je mohli splnit. Z množství dohledových systému byli nakonec vybrání tři zástupci. Jednalo se o dohledový systém Zabbix, Nagios a Zenoss (Zenoss Core) tedy o tři významné hráče na poli monitoringu. Produkty byly dále posuzovány z hlediska plnění stanovených požadavků jak při experimentální instalaci tak vzhledem ke svým uvedeným specifikacím.

3.1 Zabbix

Zabbix[**?**] je v mnohých diskuzích označován jako perfektní monitorovací software, který přináší plnohodnotný systém pro testování dostupnosti s přehledným grafickým výstupem. Svým webovým rozhraním v mnohých ohledech silně předstihoval své dva zmíněné konkurenty. V prostředí Debian Squeeze GNU/Linux proběhla jeho instalace pomocí připravených balíčků ze standardních repositářů bez potíží. Nainstalovaný software ve verzi 1.6.5-1 byl tak snadno připraven k použití. Ovládání přes webové rozhraní bylo svižné a v rámci možností i intuitivní.

Zabbix si jednoduše poradil s monitorováním dostupnosti serverů a všech "standardních" služeb. V přehledných výstupech byl snadno k nalezení jak aktuální stav tak i historie chování sledovaných služeb. Poněkud obtížněji, a snad i ve výlučném režimu ke grafickému výstupu v podobě grafů, se získávaly podrobnější informace o monitorovaných komponentách. Z pohledu možností informování uživatele Zabbix bez problému zvládnul odesílání e-mailových zpráv. V případě IM zpráv byla situace o něco horší. Pasovat ho do role XMPP klienta nebylo v porovnání s ostatními nastaveními vůbec jednoduché. I tato služba však nakonec spolehlivě fungovala. Splněny byly také požadavky na open-source software. Zabbix lze rovněž podle specifikace provozovat jako distribuovaný systém. Tato skutečnost však nebyla ověřena. V krátkodobém testovacím provozu se také zdál být stabilním a spolehlivým řešením.

Celý monitorovací systém by byl také teoreticky schopen napojení na automatizované generování konfiguračních souborů. Generovaný výstup by však musel odpovídat očekávané struktuře XML dokumentu, který je Zabbix jako jediný schopen importovat. Samotná konfigurace je následně uložená v databázi a tedy obtížně modifikovatelná mimo vlastní webové rozhraní. Plná automatizace jeho konfigurace je tak prakticky nerealizovatelná. Jako podstatný problém by se dala chápat obtížná integrace vlastních komplexnějších testovacích modulů. Při prvním seznámení se zdála být jediná cesta přímý zásah do zdrojových kódů programu.

3.2 Nagios

Dohledový systém Nagios[**?**] je v současné době označován jako jakýsi "průmyslový standard", který se zaměřuje především na testování dostupnosti. Jeho webové rozhraní poskytuje jen minimální možnosti konfigurace. Rovněž pro sběr a následnou reprezentaci sesbíraných dat například do podoby grafu je zapotřebí použití externího nástroje. Při bližším seznámení však skýtá snadno proniknutelný systém konfiguračních souborů s nepřebernými možnostmi rozšiřování své funkcionality. Jeho instalace v prostředí virtualizovaného serveru s operačním systémem Debian Squeeze GNU/Linux může proběhnout pomocí připravených balíčků, které nainstalují Nagios verzi 3.0.6. Na rozdíl od předchozího software je však dobré dopředu promyslet postup instalace a to zejména v oblasti přídavných rozšíření.

I tento dohledový software si vcelku snadno poradil jak s ověřováním dostupností zařízení pomocí počítačové sítě tak, především díky množství rozšíření, i s monitorováním většiny dnes běžně používaných síťových služeb. Konfigurace zde sice nebyla ve srovnání se softwarem Zabbix tak intuitivní. Jednalo se totiž povětšinou o přímou editaci konfiguračních souborů. Po seznámení se s jejich strukturou však nebylo zahájení monitorování nové služby nijak složitou operací. Uložení konfigurace v souborech s ne příliš složitou strukturou namísto záznamů v databázi bylo chápáno také jako přednost a to zejména kvůli jejich snadnému

[1]Seznam dohledových systémů, které byly ve fázi výběru zvažovány bude uveden v kapitole 3.

[2]Detailní popis monitorovaných komponent včetně způsobu realizace dohledu bude uveden v kapitolách níže.

generování. Upozornění uživatele na výpadek služby pomocí e-mailových či IM zpráv rovněž vyžadovalo dodatečnou konfiguraci. Ani zde ale nebyl problém se správnou funkčností či individuálním nastavením upozornění. Instalované řešení nemělo problém se stabilitou a spolehlivostí.

Předností systému Nagios bylo bezpochyby jednoduché vytváření testovacích modulů, jimiž byla jejich standardní sada rozšířena nejdříve o pokusný a po nutných úpravách produkční modul pro testování správné funkcionality komponenty mazacího serveru Virtlabu, který byl již v testovacím provozu implementována[3].

3.3 Zenoss

Zenoss[**?**] ve své plné verzi představuje komerční aplikaci, která podobně jako Zabbix poskytuje dohledový systém s přehledným grafickým výstupem a webovým rozhraním umožňujícím plnohodnotnou konfiguraci celého systému. Pomocí rozšíření, které umí přebírat například ze sytému Nagios, lze jeho funkcionalitu dále rozšiřovat. V rámci projektu je udržována také volně dostupná verze Zenoss Core, která je ovšem v mnohém omezená. Uživatel jejím použitím také přichází o podporu, která je v případě komerční verze poskytována.

Zenoss můžeme relativně snadno nainstalovat rovněž pomocí připraveného balíčku, který ovšem není součástí standardních repositářů. Je však snadno dostupný přímo ze stránek projektu. Samotná instalace není nijak složitá. Instalovaný balíček zkontroluje své závislosti na nutných komponentách, které musí být předinstalovány. Po nainstalování balíčku stručný průvodce, jenž automaticky uvítá uživatele při prvním přístupu k webovému rozhraní systému, dokončí instalační proces. Právě přes webové rozhraní je následně prováděna veškerá konfigurace. Jeho samotný vzhled a uspořádání budí sice profesionální dojem, místy je však silně neintuitivní a dle subjektivního názoru také nepřehledné.

I tento systém si poměrně snadno poradil se standardními službami. Zdůrazněme, že ve výchozím stavu tj. bez integrace dalších rozšíření počítá s prováděním monitorování jen na bázi SNMP protokolu. Zatímco vkládání připravených rozšíření v podobě ZenPack balíčků, které rozšiřují možnosti monitoringu či realizují zasílání upozornění pomocí protokolu XMPP, je vcelku jednoduchou záležitostí, import testovacího modulu pro Nagios může být přímo noční můrou. Navíc během krátkého testování importovaný modul několikrát chybně ohlásil výpadek služby, případně na změnu jejího stavu nereagoval. Zenoss Core také pravděpodobně neumožňuje import konfigurace, která je jinak uložená v databázi. Byt' systém Zenoss přináší rovněž robustní řešení, které lze rozšiřovat o další funkcionalitu, nejednalo se v případě experimentální instalace o spolehlivé a stabilní řešení.

I přes krátké praktické seznámení se s uvedenými systémy ve virtualizovaném prostředí Virtlabu bylo obtížné určit jednoznačného vítěze. Každý z nich přinášel množství výhod před ostatními s tím, že splnění vytyčených požadavků bylo vždy možné s jistou mírou dodatečného snažení. Nakonec tak byl jako nejistý vítěz vybrán dohledový systém Nagios, který oproti svým konkurentům umožňoval snadné rozšiřování jak v oblasti funkcionality vlastního systému tak v oblasti tvorby nových testovacích modulů. Následující kapitoly popisují jeho pilotní instalaci a částečně vyvinutý testovací modul pro účely použití ve Virtlabu.

[3]Vyvíjené moduly včetně principu jejich fungování budou představeny níže.

4. NASAZENÍ

Nasazení dohledového systému Nagios pro účely monitorování stavu Virtlabu probíhalo ve dvou fázích. První fáze zahrnovala instalaci virtuálního serveru s operačním systémem Debian 5.0 GNU/Linux, na němž byl nainstalován Nagios 3.0.6. Server měl být používán pro testování scénářů nasazení na Virtualizovaném prostředí Virtlabu před puštěním do ostrého provozu a především pro vývoj vlastních testovacích modulů určených pro testy specializovaných komponent. Kromě samotného Nagios 3 instalace serveru dále zahrnovala

- Nagios NRPE plugin - podpora vzdáleného vykonávání testovacích modulů
- Nagios plugins - obecná sada testovacích modulů pro Nagios
- Nagios SNMP plugins - sada nástrojů pro testování zařízení pomocí SNMP
- NDOUtils Nagios3 MySQL - podpora ukládání událostí Nagios 3 do databáze MySQL
- XMPP utils - podpora komunikace pomocí protokolu XMPP

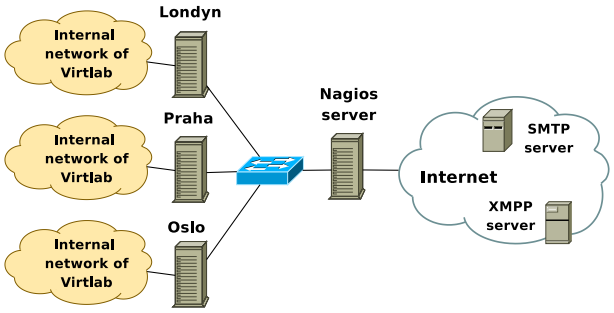
Při instalaci byly primárně využívány standardní balíčky obsažené v repositářích projektu Debian GNU/Linux. Prakticky celá instalace tak mohla být provedená jediným příkazem

```
aptitude install nagios3 nagios-nrpe-plugin nagios-plugins
nagios-plugins-standard nagios-plugins-basic nagios-snmp-
plugins ndoutils-nagios3-mysql
```

Přidání podpory XMPP protokolu bylo realizováno manuálním vložením rozšíření sekcí *host-notify-by-jabber* a *notify-by-jabber* v konfiguračním souboru *commands.cfg* následované vytvořením obslužného skriptu, na který je z obou sekcí odkazováno. K tomuto účelu byl použit a otestován existující skript dostupný na [**?**].

K tomu aby mohly být na monitorovaných serverech virtualizovaného Virtlabu spouštěny vzdáleně testovací moduly systému Nagios, bylo zapotřebí jejich instalaci rozšířit o *Nagios NRPE server* a potřebné testovací moduly. I zde bylo využito primárně balíčku obsažených v repositářích distribuce Debian GNU/Linux. V případě vyvíjených testovacích modulů byla jejich distribuce na servery zajišťována ve vlastní režii.

Virtualizované prostředí tak obsahovalo server s dohledovým systémem Nagios a tři virtualizované lokality distribuovaného Virtlabu. Pro účely komunikace s okolím byl nezbytný SMTP server a server zprostředkovávající komunikaci pomocí protokolu XMPP (dříve Jabber). Schématické znázornění síťového propojení reprezentuje obrázek 1.



Obrázek 1: Schématické znázornění testovacího prostředí pro optimalizaci nasazení systému Nagios ve Virtlabu

Druhou fází instalace mělo být kompletní oddělení dohledového serveru od virtualizovaného prostředí, tedy příprava a nasazení dedikovaného serveru. Tento krok však po zvážení možných důsledků výpadku dohledového systému Virtlabu včetně zohlednění bezpečnostních aspektů virtualizačního prostředí, proveden nebyl. Nahrazení virtuálního serveru fyzickým, jenž by nabízel podobnou úroveň odolnosti proti selhání, by přineslo nepřiměřeně vysoké náklady k získaným výhodám. Následujícím krokem, místo vytvoření nového serveru, tak bylo provedení potřebných změn v instalaci a především zabezpečení virtuálního stroje. Ze serveru byly postupně odstraněny služby, které nebyly potřeba pro běh dohledového systému. Následovala také příprava a aplikace zabezpečení serveru pomocí firewallu, který na něm omezoval nežádoucí síťový provoz. V přímé závislosti na Nagios 3 vytvořený firewall obsahoval jen pravidlo umožňující navazování odchozího spojení na TCP port 5666[4], jenž je využíváno při vykonávání testu na vzdálených systémech pomocí *NRPE démona*.

4.1 Konfigurace Nagios 3

Automatizovaná instalace Nagios 3 z připravených distribučních balíčků ve svém průběhu provádí četná nastavení, jenž jsou pro správný chod nezbytná. Automaticky je rovněž provedena konfigurace webového serveru, díky níž již stačí nastavení nutné webové autentizace na obsah adresáře */etc/nagios3/*, ve kterém Nagios vedle svých konfiguračních souborů ukládá také prováděcí skripty webového rozhraní. Autentizaci lze nastavit například pomocí příkazu

```
htpasswd -c /etc/nagios3/htpasswd.users nagiosadmin
```

Uživatelské jméno *nagiosadmin* je vyžadováno z důvodu návaznosti na výchozí konfigurace. Může však být v součinnosti s dalšími změnami upraveno. Od tohoto okamžiku je webové rozhraní připraveno ke sledování a v omezené míře také konfiguraci chování dohledového systému.

Primárním cílem nasazeného systémy nebyl dohled pouze nad serverem, na němž byl provozován, ale především monitorování chování jednotlivých lokalit Virtlabu. Další konfigurace byla tedy nezbytná.

4.1.1 Upozornění uživatele

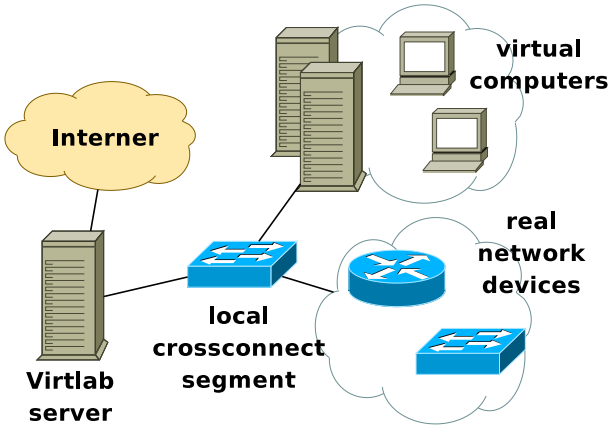
Jako následující krok konfigurace bylo zvoleno celkové nastavení upozornění osob na abnormální stavy. Výchozí metodou upozornění bylo zvoleno zasílání e-mailových zpráv. K tomu, aby tento způsob notifikace fungoval, je zapotřebí provést konfiguraci služby, která bude generované e-maily předávat příslušnému SMTP serveru. Zde bylo použito transportního agenta Postfix. Na straně systému Nagios byly následně definovány kontaktní osoby, které jsou za jednotlivé monitorované části zodpovědné. Speciální komponenty Virtlabu jsou totiž úzce svázány s konkrétními osobami, kteří se podílejí na jejich vývoji a tudíž jejich informovanost často vede k efektivnějšímu řešení problému. Mechanizmus upozornění pak fungoval následujícím způsobem

- dohledový systém mohl pro monitorovanou službu, v závislosti na jejím stavu, vyprodukovat některou z událostí upozornění, nedostupnost, kritický stav nebo obnovení činnosti služby
- vzniklá událost byla předána k přeposlání předem definované skupině příjemců, která automaticky obsahovala kontakt na administrátora serveru a případný kontakt pracovníka, odpovědného za vývoj dané komponenty
- před vlastním odesláním na adresy osob uvedených v dané skupině byly zprávy dále filtrovány podle nastavení příjemce
  - Administrátor má omezené přijímání zpráv produkované při události upozornění. Tyto zprávy mu jsou doručovány pouze jednou. O ostatních událostech a to i opakovaných vlivem přetrvávajícího problému je však informován v hodinových intervalech bez dalších omezení.
  - Odpovědný pracovník obdrží informaci o vzniklé události jen jednou.

Jako další způsob informování osob o vzniklých událostech byl zvolen komunikační protokol XMPP resp. služba gTalk. Tato volba byla provedena především z důvodu rozšíření a přednosti rychlejšího doručování zpráv aktivním klientům oproti e-mailové komunikaci. Jejím cílem nebylo nahrazení upozornění e-mailem, ale pouze vytvoření dalšího alternativního kanálu, jenž byl v testovacím provozu využíván pouze administrátorem serveru a následně po přechodu do produkčního prostředí těmi uživateli, kteří si tento způsob explicitně vyžádali.

Pro notifikaci pomocí protokolu XMPP nebyla definována žádná dodatečná omezení. Zpráva je tak příjemci odeslána bezprostředně po vzniku oznamovaného problému. Výhodou tohoto způsobu rozesílání informací oproti využití SMTP v prostředí Vysoké Školy Báňské - Technické Univerzity Ostrava, je prakticky neomezená rychlost odesílání generovaných zpráv příjemci. Použitý SMTP server totiž omezuje množství zpráv odeslaných za jednotku času. Prakticky bezlimitní možnosti služby gTalk se tak z důvodu nutnosti doručení velkého množství upozornění hodily zejména v okamžiku ladění a testování dohledového systému.

[4]Detailní obsah bezpečnostních pravidel firewallu dohledového serveru, které byly nezbytné pro jeho činnost, nebude k vůli lokální povaze diskutován. Na serveru byl firewall realizován pomocí pravidel IPtables.



Obrázek 2: Schématické znázornění interní počítačové sítě Virtlabu

4.1.2 Monitorované služby

Po dokončení konfigurace upozorňování uživatele bylo zapotřebí vydefinovat a nakonfigurovat služby, jenž mají být monitorovány. I zde byly nejdříve definovány standardní "dobře známé" služby, pro něž postačovala konfigurace modulů dohledového systému, které zajišťovaly jejich testování a byly nainstalovány v balíčku rozšíření Nagios 3. Pro specifické služby bylo zapotřebí nejprve testovací modul vyvinout a poté začlenit do dohledového systému.

Aby bylo nasazení dohledového systému na lokalitu Virtlabu co možná nejjednodušší, byly stanoveny skupiny služeb a vlastností serverů, jejichž výskyt se i v budoucích lokalitách předpokládá. Mezi ně patřila skupina stavu hlavního serveru, která zahrnovala testy jeho aktuální zátěže, využití diskového prostoru, stav disků poskytnutý S.M.A.R.T. démonem, celkového počtu procesů a počtu tzv. zombí procesů. Sledování všech zmíněných informací o stavu serveru by mělo být realizováno na každé lokalitě s tím, že jeho provedením na ní bude pověřen NRPE démon. Podobně i dostupnost hlavního serveru a funkčnost služby Secure Shell byla na všech monitorovaných lokalitách kontrolována. Bylo tedy provedeno jejich začlenění do skupiny služeb stavu serveru. Jako pomocný zdroj informací pro případy hledání příčin a řešení závažnější problémů byla tato skupina znovu rozšířena. Tentokráte přibyla kontrola funkčnosti NTP serveru, počtu přihlášených uživatelů a neodeslaných e-mailových zpráv ve frontě Postfix serveru. Události, které byly tímto seskupením generovány, obdržel vždy administrátor.

Kromě hlavního serveru bylo zapotřebí realizovat dohled vybavení, které bylo obsaženo v interní počítačové síti lokalit. Jednalo se především o podpůrné servery, které pomocí virtualizace poskytují simulované počítače ve vybavení Virtlabu a dále například o přepínací prvky segmentu virtuálního spojovacího pole[5]. Obrázek 2 reprezentuje strukturu interní počítačové sítě lokalit.

Pro monitorování prvků a serverů v interní síti lokalit Virtlabu, vzhledem k různému počtu a specifičnosti každého z nich v různých lokalitách, nebylo možno vytvořit jednotnou dostatečně obecnou skupinu, jenž by mohla být aplikována na všechny sledované lokality. Vznikly tak skupiny sledovaných komponent pro jednotlivé lokality, které odrážely reálný počet daného vybavení v každé z nich. I zde bylo zapotřebí využít *NRPE démona* na hlavním serveru, neboť interní síť není jinak než přes hlavní server dostupná. Na prvcích interní sítě bylo ověřování jejich funkčnosti a sledování zátěže prováděno pomocí SNMP dotazů zasílaných z hlavního serveru. Stejně tak i zde obdržel upozornění na vzniklé události administrátor serveru.

První specializovaná komponenta Virtlabu, jejíž úplný dohled byl realizován, je tzv. mazací server. Jeho úkolem je odstranění uživatelem provedených konfigurací v době jeho rezervace a navrácení zařízení do výchozího definovaného stavu[6]. Pro jeho dohled vznikla samostatná skupina monitorovacích pravidel, která zahrnovala kontrolu přítomnosti procesu mazacího serveru, dostupnost jeho management rozhraní a ověřování jeho funkcionality. Zatímco kontrola přítomnosti jeho procesu na serveru byla provedena pomocí standardního testovacího modulu, kontroly dostupnosti jeho managementu a funkcionality vyžadovaly vlastní kontrolní modul. Vyvinutý modul realizoval kontrolu dostupnosti management rozhraní, ověřoval celkový počet zařízení, které je mazací server schopen obsloužit a sledoval přítomnost speciálního testovacího zařízení v jejich interním seznamu mazacího serveru. Testovací zařízení představovalo samostatnou část vyvinutého modulu, na které mohl server provést pomyslné "smazání" její konfigurace. Test spočíval v navázání TCP spojení na příslušný port komponenty mazacího serveru a následné zaslání testovacích sekvencí. V závislosti na úspěšnosti vyhodnocení zpětného výpisu a výsledku "mazání" pomyslné konfigurace v části testovací komponenty, byl navrácen

- bezchybný stav - při vyhodnocování zpětného výpisu byl v interním seznamu zařízení mazacího serveru nalezen konfigurovaný počet záznamů, který navíc obsahoval testovací zařízené. Rovněž smazání testovacího zařízení proběhlo bezchybně.
- upozornění - v interním seznamu nebyl nalezen konfigurovaný počet záznamu. Zbylé testy však proběhly úspěšně.
- kritický stav - v případě nezdaru při odstraňování konfigurace z testovacího zařízení nebo jeho absenci v interním seznamu serveru. Také samozřejmě v případě nedostupnosti management konzole.

Dohledový systém na základě navrácené hodnoty mohl vygenerovat případnou událost a rozeslat ji kontaktní skupině mazacího serveru.

Pro komponenty konzolového, tunelovacího a rezervačního serveru byly rovněž vydefinovány, především z důvodu snadné integrace budoucích testovacích modulů, skupiny kontrolních pravidel. Během psaní tohoto příspěvku však nebyly hotovy všechny speciální testovací moduly. Monitorována tak

[5]Význam Virtuálního spojovacího pole je detailně vysvětlen v [?]. Jeho účelem je realizace propojení síťových komponent uživatelem požadované topologie

[6]Komplexnější popis komponenty mazacího serveru naleznete na [?]

byla pouze přítomnost a počet procesů na serveru rozšířená o dostupnost a kontrolu stavu jejich management konzole. Vyvíjené testovací moduly totiž vyžadují mnohem hlubší provázání s interními strukturami Virtlabu, kontrolu aktuálně komponentami prováděných operací v závislosti na počínání přihlášených uživatelů či vzniklých rezervací. Jejich vývoj bude dále realizován formou bakalářské práce. Ve stejném stádiu rozpracovanosti je i speciální modul pro kontrolu spojovacího zařízení ASSSK, které bylo vyvinuto na Vysoké Škole Báňské - Technické Univerzitě Ostrava a slouží k automatizovanému propojování sériových linek mezi směrovači.

Jako předchozí komponenty i pro ověřování funkčnosti webového serveru vznikla skupina testovacích pravidel, jenž zahrnovala ověření funkčnosti webového serveru, ověřování platnosti serverového certifikátu a jednoduché ověření obsahu poskytnutého serverem[7]. Dodatečně bylo také do této skupiny začleněno monitorování vlastního procesu Apache2 a sledování dostupnosti MySQL serveru, které jsou ve Virtlabu pro tvorbu webových stránek používány. Případné produkované události obdrželi vedle administrátora také tvůrci webového rozhraní.

5. TESTOVÁNÍ

Finální konfigurace dohledového systému byla především ve virtualizovaném prostředí podrobena řadě testů, které měly odkrýt její slabá místa. Z důvodu průběžného testování jednotlivých dohledových schémat již v okamžicích jejich plánování, nebylo při výsledné kontrole nalezeno mnoho problémů, které by monitorovací systém nedokázal detekovat. Významnější nedostatky byly nalezeny v ověřovacích pravidlech konzolového serveru, kdy jeho nevhodnou konfigurací a záměrnou manipulací s prostředky serveru byl nasimulován případ falešně pozitivního výsledku testu a to i přes to, že konzolový server nebyl nadále schopen plnit svou funkci. Tento problém bude pravděpodobně odstraněn nasazením nového testovacího modulu, jehož vývoj by měl být dokončen v únoru roku 2010.

Vedle konzolového serveru se testování zaměřilo také na ostatní specializované komponenty a samozřejmě samotné hlavní servery lokalit. Na základě analýzy výsledků testu bylo provedeno pomyslné první kolo optimalizace parametru dohledových pravidel. Druhé kolo optimalizací pravděpodobně vyplyne z dlouhodobějšího reálného provozu. Z tohoto důvodu předpokládáme nejméně čtyř měsíční součinnost obou monitorovacích systémů, tedy jak integrovaného monitorování komponent, tak samotného systému Nagios.

6. ZÁVěR

Nasazovaný dohledový systém v současné době opustil fázi příprav a testování schémat nasazení v distribuované Virtuální laboratoří počítačových sítí. Úspěšně bylo zakončeno základní testování v podmínkách virtualizovaného Virtlabu. Práce na nasazení dohledového systému v produkčním prostředí jsou rovněž na lokalitě VŠB-TU Ostrava hotovy. Podle předběžných výsledků, které poskytlo testování systému ve virtualizovaném prostředí a následně prvních výsledků produkčního provozu v rámci lokality VŠB-TU Ostrava, se systém zdá být plně funkční, spolehlivou a pružnou náhradou za doposud používaný integrovaný monitoring přímo v hlavních serverech lokalit. Předpokládáme, že jeho nasazení napomůže k další stabilizaci prostředí Virtlabu a to předcházením a včasnou detekcí případných anomálií. V rámci dalšího rozvoje je také plánována integrace nových komplexnějších testovacích modulů, které bezpochyby dále zkvalitní služby dohledu.

[7]V terminologii Virtlabu je často sada PHP skriptů, která realizuje jak přístup uživatelů k systému Virtlabu tak sofistikované operace nutné při provádění rezervací, označována jako řídící server. Více o této komponentě na [**?**].