Problémy nasazení distribuovaného e-learningového systému virtuální laboratoře počítačových sítí do praktického provozu

Petr Grygárek *
petr.grygarek@vsb.cz

Tomáš Kučera*
tomas.kucera@vsb.cz

Jan Vavříček*
jan.vavricek@vsb.cz

Abstrakt Na katedře informatiky FEI VŠB-TU již několik let vyvíjíme architektury pro vzdálené zpřístupnění laboratoře počítačových sítí pomocí Internetu. V roce 2007 jsme zprovoznili distribuovanou verzi mezi VŠB-TU v Ostravě a OPF Slezské univerzity v Karviné, která umožňuje vytvářet distribuované virtuální topologie a sdílet laboratorní vybavení obou těchto lokalit. Příspěvek popisuje problémy a zkušenosti z prvních měsíců nasazování a reálného provozu a poukazuje na možná řešení, která lze aplikovat při provozování mnohých dalších distribuovaných e-learningových systémů.

Klíčová slova: virtuální laboratoř, počítačové sítě, distribuovaný e-learningový systém

1 Úvod

Koncem roku 2007 byla mezi FEI VŠB-Technické univerzity Ostrava a Obchodně-podnikatelskou fakultou Slezské univerzity v Karviné zprovozněna pilotní konfigurace virtuální laboratoře počítačových sítí [1], založená na plně distribuované architektuře [2]. Tento distribuovaný e-learningový systém nazvaný Virtlab umožňuje vzájemně sdílet laboratorní zařízení rozmístěné v několika lokalitách propojených prostřednictvím Internetu a automaticky toto zařízení propojovat do virtuálních topologií potřebných pro úlohy rezervované pro vzdálené řešení studenty. Laboratorní zařízení použitelná k sestavení konkrétní úlohy na dobu časového okna rezervovaného studentem jsou v době rezervace vyhledávána dynamicky ve všech lokalitách.

Distribuovaná virtuální laboratoř byla vybudována na základě několikaletých zkušeností s lokálními řešeními vzdáleného zpřístupnění laboratorních zařízení ([3],[4]) a prototypy hardwarových zařízení a architektur pro automatizované propojování lokálních i distribuovaných síťových topologií ([5]). Přestože jsme během provozu předchozích lokálních řešení získali mnohé zkušenosti se zajištěním trvalého provozu systému použitého jako součást výuky studentů v několika předmětech, při nasazování distribuované verze jsme byli nuceni řešit problémy i zcela jiného charakteru, které plynuly z distribuované povahy, výrazně většího rozsahu i různých organizačních zvyklostí participujících institucí a částečně konfliktních požadavků na časový plán využití laboratorního vybavení s ohledem na lokální rozvrhy výuky. Jelikož mnohé ze získaných zkušeností považujeme za užitečné

* Vysoká škola báňská – Technická univerzita Ostrava, kat. informatiky FEI, 17. listopadu, 708 33 Ostrava-Poruba

pro organizaci reálného provozu mnohých jiných distribuovaných e-learningového systémů, shrnuli jsme nejdůležitější z nich v následujícím příspěvku.

2 Problémy distribuovaného systému

Mnohá opatření, která se ukázalo potřebné realizovat, byla nutná již ze samotného distribuovaného charakteru budovaného systému. Přestože lze konstatovat, že plně distribuovaná architektura Virtlabu se v praxi velmi osvědčila a při rozpadu komunikace mezi lokalitami dává možnost zachování samostatné funkce jednotlivých lokalit nebo jejich propojených skupin, je plná funkčnost systému samozřejmě zcela závislá na komunikační infrastruktuře. Jelikož rozpad komunikace mezi lokalitami se projeví nejen v omezení množiny laboratorních zařízení, které jsou k dispozici pro rezervace uživatelů, ale i v nemožnosti přistoupit k dříve rezervovaným zařízením, byli jsme nuceni vybudovat vlastní systém monitoringu konektivity mezi lokalitami. Monitorovací systém byl postaven na bázi open-source software Zabbix [6], který umožňuje archivovat informace o časových intervalech rozpadu konektivity a pomocí e-mailu informovat správce o těchto kritických událostech. Přestože není správa síťové infrastruktury v kompetenci správců Virtlabu, pro řešení případných chyb nahlášených uživateli systému je velmi důležité mít možnost zpětně dohledat, zda interakce mezi komponentami systému neproběhla správně z důvodu implementační chyby, nebo dočasného výpadku konektivity.

Zkušenosti z provozu ukázaly, že hlavním problémem konektivity nejsou ani tak dlouhodobější výpadky dostupnosti lokalit, které virtuální laboratoř řeší svou plně distribuovanou povahou, ale spíše nečekané krátkodobé výpadky spojené např. s aktuálním počasím v případě lokalit připojených optickým pojítkem citlivým na mlhu nebo intenzivní déšť. Jelikož těmto problémům nelze bez větších investic do síťové infrastruktury nijak předejít, ukázalo se výhodné uživatele provádějícího rezervaci nebo požadujícího přístup na konzole laboratorních prvků nedostupné lokality o vzniklé situaci alespoň informovat. Toho lze dosáhnout nalinkováním stránky monitorovacího systému z GUI Virtlabu. V souvislosti s tím se ukázalo potřebné poskytnout uživateli informaci, ve kterých lokalitách jsou umístěny jednotlivé fyzické laboratorní prvky použité pro realizaci jeho distribuované virtuální topologie. Přestože informace o fyzickém umístění laboratorních prvků byla doposud před uživatelem s ohledem na filosofii plné transparence distribuované povahy systému úmyslně zcela skryta, ukázala se pro mnohé uživatele i bez ohledu na případné dohledání příčiny výpadku překvapivě atraktivní.

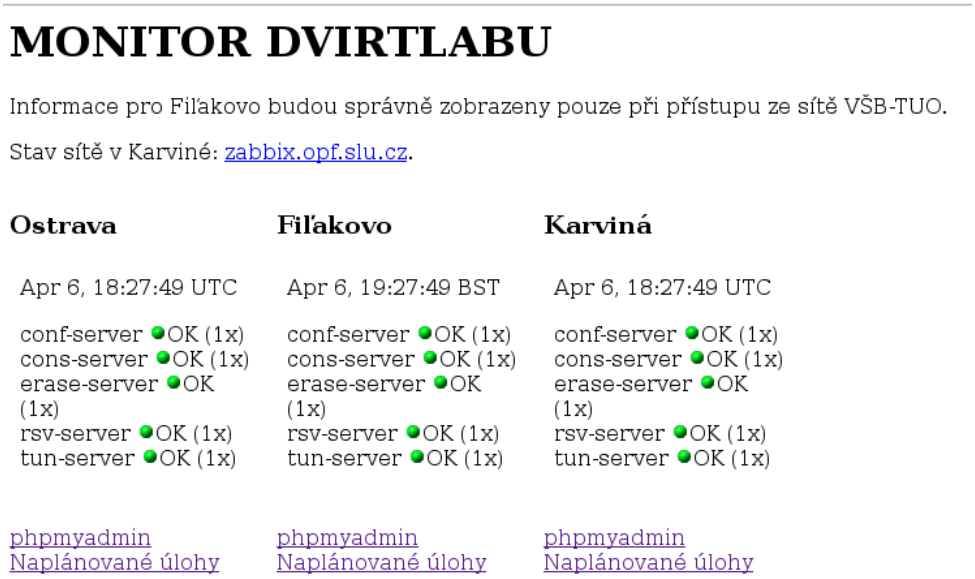
2.1 Záznam o chodu systému a interakcích mezi komponentami

Pro identifikaci zdrojů chyb se také ukázal nezbytný dobře promyšlený systém logování akcí, které v systému proběhly. Všechny komponenty proto jednotně logují na Syslog server své lokality s tím, že je navržen systém úrovní významnosti zaznamenávaných událostí. Tak je možné na jedné straně snadno odfiltrovávat pouze informace o kritických chybách a reagovat na ně zasláním e-mailové zprávy příslušnému administrátorovi a na druhé straně detailně sledovat chod systému při spuštění komponent v režimu logování hlášení v úrovní Debug. Bylo vypracováno unifikované API pro logování použitelné ve všech jazycích, v nichž jsou komponenty Virtlabu napsány (C, PHP, Perl, Bash skripty) [7]. Rovněž se osvědčilo ve zprávách předávaných mezi komponentami distribuovaného systému předávat identifikátor transakce, kterým jsou označeny všechny zprávy vázané vzájemnou

příčinnou souvislostí. Tím, že se identifikátor transakce loguje spolu s informací o přijaté zprávě, je poměrně snadné vysledovat souvislost událostí i v systému složeném z většího množství komponent distribuovaných v několika lokalitách a zaznamenávajících zprávy o chodu na různé Syslog servery.

2.2 Podpora rychlé diagnostiky chyb

Pro rychlé nalezení příčiny zjištěných chyb funkce systému se ukázalo vhodné také implementovat systém, který monitoruje běh procesů jednotlivých komponent systému ve všech lokalitách. K tomu účelu jsme vybudovali správcovský portál dostupný oprávněným uživatelů [8]. V současné době jsou komponenty monitorovány pouze na úrovni běhu příslušných procesů, plánujeme však implementaci testů správné reakce komponent na požadavky klientů s použitím systému Zabbix a vhodně implementovaných externích testovacích skriptů (external checks). Ukázka vzhledu monitorovací aplikace je na obr. 1.



Obrázek 1: Ukázka z GUI monitorovací aplikace

Pro rychlou lokalizaci závad se rovněž osvědčilo odkazem ze správcovského portálu přehledným způsobem zpřístupnit obsah databází jednotlivých lokalit. K tomu se ukázal velmi vhodný volně dostupný software PHPMyEdit [9]. Pomocí vlastní WWW aplikace zpřístupňujeme také další systémové informace z jednotlivých lokalit, zejména seznam akcí naplánovaných pro aktivaci a deaktivaci virtuálních topologií před a po jednotlivých uživateli rezervovaných časových úsecích.

S ohledem na to, že je virtuální laboratoř nasazena do výuky formou úloh povinně vypracovávaných k získání zápočtu, bylo nezbytné také zorganizovat systém ohlašování chyb a nedostatků zjištěných uživateli a zajistit reakci zejména na kritické chyby v definovaném čase. Právě tato potřeba se ukázala v akademickém prostředí postrádajícím prostředky na rutinní provoz takovýchto systémů jako nejvíce problematická. Nejprve bylo třeba vybudovat vhodný systém pro ohlašování chyb a sledování stavu jejich odstranění. Po krátkodobém provozování existujících systému Bugzilla [10] a Mantis [11] jsme se rozhodli pro

implementaci vlastního systému VirtIS [12], který lépe vyhovuje akademickému prostředí s týmem dobrovolně pracujících vývojářů než komerčněji zaměřené systémy s pevnější organizací procesů. Také bylo třeba uživatele motivovat k ohlašování chyb v takové formě, která povede k rychlému dohledání příčiny a sjednání nápravy. Proto jsme formulář pro ohlašování chyb integrovali přímo do GUI portálů virtuální laboratoře ve všech lokalitách a jeho výstup provázali se systémem VirtIS. Mimo nutnosti specifikace času výskytu chyby se jako velmi přínosná ukázala možnost vložit k popisu chyby obrázek, nejčastěji snímek obrazovky dokumentující nesprávné nebo neočekávané chování systému.

3 Stanovení časového plánu sdílení laboratorních zařízení

Přestože participující instituce mají zájem využít své laboratorní vybavení co nejefektivněji a nabídnout je v době malého využití partnerským lokalitám, je zřejmé, že při reálném začlenění virtuální laboratoře do výuky bude primárním zájmem každé instituce zajistit dostatečný přístup k laboratorním prvkům zejména pro své vlastní studenty. Proto se ukázalo nezbytným dát správcům lokalit k dispozici nástroj, kterým budou moci u každého laboratorního zařízení stanovit, kdy bude laboratorní zařízení k dispozici uživatelům ze kterých lokalit. Tento nástroj musí být dostatečně flexibilní, aby bylo v případném budoucím začlenění lokalit neakademických institucí možné postihnout nejrůznější dohody a omezující podmínky pro zápůjčky prvků mezi lokalitami.

Časový rozvrh nabídky jednotlivých laboratorních prvků každé partnerské lokalitě konfiguruje správce lokality v konfiguračním souboru svého rezervačního serveru, tj. komponenty, která vyřizuje a eviduje žádosti o zápůjčky laboratorních zařízení lokality od řídících serverů ostatních i své vlastní lokality. Jako základ jsme zvolili konfiguraci týdenního časového rozvrhu, který přiřazujeme jednotlivým zařízením. Tam je možné nadefinovat hodiny v rámci jednotlivých dnů, v nichž bude zařízení zapůjčováno. Pro každou partnerskou lokalitu pak správce pro jednotlivá lokální zařízení specifikuje, podle kterého z týdenních časových rozvrhů bude dané zařízení dané partnerské lokalitě k dispozici. V současné době jsou všechna zařízení obou lokalit k dispozici až na servisní okna prakticky neustále, předpokládá se však, že pokud začneme virtuální laboratoř využívat i v rámci prezenční výuky během laboratorních cvičení, budou správci lokalit týdenní rozvrhy měnit typicky před začátkem každého semestru tak, aby v době využití v prezenční výuce byly lokální laboratorní prvky vyhrazeny pro rezervace uživatelů vlastní lokality. V praktickém provozu se osvědčilo stanovit v rozvrhu nabízení laboratorních prvků okna pro servis zařízení sjednocená ve všech lokalitách a také nadefinovat „prázdný“ časový rozvrh, který může být kdykoli snadno přiřazen porouchanému zařízení nebo zařízení, na kterém bude probíhat dlouhodobější údržba a je nutné jej ze systému dočasně odstavit.

4 Správa uživatelských účtů

Dalším z rysů systému Virtlab, který se v reálném provozu velmi osvědčil, je plně distribuovaná správa uživatelských účtů, kdy každá lokalita provozuje svou vlastní databázi uživatelů. Za správu účtů uživatelů lokality jsou tedy zodpovědní správcové příslušné lokality. Lokální ověřování uživatelů portálem jejich domovské lokality také usnadňuje integraci s rozličnými lokálními autentizačními systémy lokalit, což omezuje počet hesel, které si musí uživatelé pamatovat. V lokalitách VŠB FEI Ostrava i OPF SLU Karviná

jsou v pilotní konfiguraci hesla uživatelů ověřována proti LDAP; zvažována je také implementace možnosti ověřování proti v současné době postupně budovaným globálním autentizačním infrastrukturám, zejména EduRoam [13] a Shibboleth [14]. Pro provoz nezbytnou se ukázala i možnost současného použití staticky vložených hesel uložených přímo v databázi uživatelů, nezbytná pro externí uživatele, kteří nejsou zavedení v LDAP žádné spolupracující instituce.

Podstatnou výhodou lokálních databází uživatelů je, že správcům lokalit může být dán přístup do databáze i na úrovni systému. To umožňuje tvorbu pro lokalitu specifických utilit použitelných pro import uživatelů přímým vložením do lokální databáze s využitím SQL příkazů, užitečných pro vkládání seznamů uživatelů exportovaných z nejrůznějších databází uživatelů provozovaných jednotlivými institucemi.

Velmi praktickým se ukázalo rovněž přiřazování uživatelů do skupin a možnost určení data expirace uživatelského účtu při jeho vytváření. Skupiny uživatelů jsou výhodné zejména s ohledem na zpřístupnění virtuální laboratoře studentům některých předmětů pouze na dobu jejich studia – účty, jimž skončila doba platnosti, a účty studentů skupin, které již studium předmětů využívající virtuální laboratoř ukončily, mohou být snadno promazány. Do budoucna také zvažujeme možnost vyhrazení časových úseků (stanovených jako periodicky opakované nebo jednorázové) pro rezervace uživatelů určitých skupin.

4.1 Uživatelské kvóty

Pro zajištění férového přístupu k zařízením a omezení nekorektního chování nedisciplinovaných uživatelů se ukázala nutnou implementace vhodného systému kvót, který bude omezovat množství rezervací jednotlivých uživatelů.

Prvním implementovaným řešením bylo periodické nastavování kvóty pro rezervace každému uživateli vždy na začátku kalendářního týdne. Uživatel tak mohl rezervovat časová okna až do vyčerpání této každotýdenně přidělované kvóty. Jelikož však nebylo bráno v úvahu, na kdy uživatel svou rezervaci naplánoval, docházelo v některých případech k nežádoucímu „spoření“ kvót: uživatel si několik týdnů v rámci periodicky obnovovaných kvót postupně rezervoval časové úseky jistého období v budoucnu, ve kterém si pak využití virtuální laboratoře prakticky monopolizoval.

Další úvahou bylo přidělovat každému uživateli určité množství hodin na jednotlivé kalendářní týdny v budoucnosti. Implementace tohoto mechanismu se však nejevila příliš pružnou s ohledem na případnou potřebu rozšíření jednotkového intervalu pro sledování kvóty nad jeden týden. Převody dat v budoucnosti na dny v týdnu a vymezení hranic týdnů, do kterých zasahují dlouhodobější rezervace navíc v některých použitých programovacích jazycích bez příslušných knihoven přinášely nepřiměřené implementační problémy.

Jako nejvýhodnější systém kvót se v praxi ukázal mechanismus plovoucího časového intervalu, během něhož se množství hodin rezervovaných uživatelem vyhodnocuje. Při každém požadavku na rezervaci se pouze zjistí, zda uživatel nepřekročí celkovým počet rezervovaných hodin v časovém okně o celo-systémově přednastavené délce, která je umístěna v čase symetricky ke středu časového intervalu zamýšlené rezervace. Každý uživatel má nastaven celkový počet hodin, které může v (libovolně umístěném) plovoucí okně pevné délky čerpat. Výhodou je, že není třeba sledovat aktuální hodnotu zbývající kvóty jednotlivých uživatelů ani periodicky navyšovat hodnotu jejich kreditu. Navíc kvóta nemusí být zpětně navyšována při případném rušení rezervací. Musí se pouze ošetřit, aby

uživatel nemohl zrušit své již proběhnuvší rezervace, čímž by si mohl uměle navyšovat kvótu (v nahlíženém okně by systém evidoval u uživatele méně hodin, než ve skutečnosti virtuální laboratoř využíval).

Podobně jsme zatím uživateli zamezili rušit rezervace, jejíž časový interval právě probíhá. Do budoucna plánujeme dát uživateli možnost rezervaci zkrátit v případě, že s úlohou skončí dříve, než původně odhadoval. V systému kvót se mu pak započítají pouze skutečně spotřebované hodiny a laboratorní zařízení nebude zbytečně blokováno až do času původně zamýšleného konce rezervace.

V systému kvót bylo také třeba zohlednit fakt, že na úloze může zároveň spolupracovat více studentů. V současné době je čas odečítán každému z uživatelů, kteří na úloze pracují. Započítávat tento čas pouze studentovi, který je primárním vlastníkem dané rezervace, by dávalo uživateli příležitost po vyčerpání své kvóty požádat některého ze svých kolegů s dostatečnou kvótou o rezervování úlohy, kde by vystupoval pouze jako spoluřešitel.

5 Generátor konfiguračních souborů

Jistou daní za flexibilitu systému distribuované virtuální laboratoře je jeho poměrně komplexní konfigurace pomocí sady vzájemně provázaných textových souborů distribuovaných v jednotlivých lokalitách. Část této konfigurace má globální charakter a je identická pro všechny lokality (seznam adres systémových komponent ve spolupracujících lokalitách), část je naopak specifická pro jednotlivé lokality a ostatním lokalitám přístupná pouze nepřímo. Informací specifickou pro každou lokalitu je zejména výčet a parametry lokálního laboratorního vybavení včetně popisu jeho připojení k distribuovanému virtuálnímu spojovacímu poli a časový rozvrh nabídky lokálního vybavení pro rezervace z jiných lokalit.

Nezbytná je také správnost popisu přemostění spojení příchozích od vzdáleného emulátoru terminálu z GUI uživatele na RS232 konzole a virtuální terminály všech laboratorních zařízení lokality a nastavení parametrů pro skripty, které mažou konfigurace laboratorních prvků použitých v rezervované topologii před jejich zpřístupněním uživateli.

Konzistence globální části konfigurace je pro funkčnost systému velmi kritická. Chybné nakonfigurování adres rezervačních serverů vede k tomu, že u dotčených lokalit není možné vyjednávat zápůjčky laboratorních prvků. Při nesprávném nastavení adres konzolových serverů nejsou uživateli zpřístupněny management rozhraní reálných ani simulovaných síťových prvků příslušných lokalit. Pokud jsou nesprávně nastaveny adresy konfiguračních a mazacích serverů, nezdaří se žádost o propojení požadované distribuované virtuální topologie a vymazání konfigurace laboratorních zařízení v okamžiku začátku časového úseku pro řešení dříve rezervované úlohy.

Praktická zkušenost z úvodní fáze budování pilotní konfigurace Ostrava-Karviná ukázala, že přes 75% závažnějších chyb systému bylo způsobeno nikoli chybou v implementaci, ale nekonzistentní konfigurací v některé z lokalit.

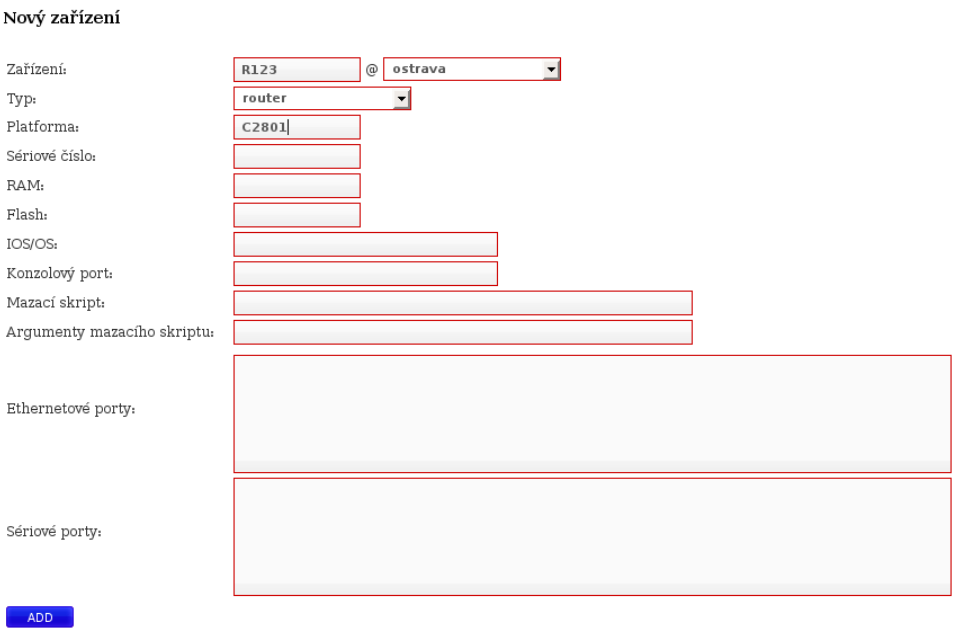
Pro zajištění konzistence konfigurace a dosažení přehledu o konfiguraci všech částí distribuovaného systému nutného při odhalování provozních chyb se ukázalo užitečné vytvořit centralizovanou webovou aplikaci [15], která udržuje globální informaci o konfiguraci systému ve své databázi a konzistentní konfigurační soubory generuje automaticky. Filosofie plně distribuované architektury tím není nijak narušena; jedná se pouze o pomocný nástroj pro konfiguraci. Instalace automaticky vytvořených konfiguračních souborů na servery své lokality zůstává i nadále v kompetenci lokálního správce, který pro ní má k dispozici vhodné systémové nástroje. Operace prováděné v konfigurační aplikaci jsou

chráněny systémem uživatelských práv, kdy pouze správcové jednotlivých lokalit mají právo modifikovat lokální konfiguraci, zatímco přidávat či odebírat lokality smí výhradně globální správce systému. Jelikož udržování databáze v souladu s aktuálním stavem laboratorního vybavení lokality je pro správce lokalit nezbytné pro automatické generování správných konfiguračních souborů, získali jsme rovněž zdroj dat, z něhož lze kdykoli vygenerovat aktuální dokumentaci o stavu celého systému, aniž bychom správce nutili při každé lokální změně aktualizovat lokální dokumentaci.

Globální informace o konfiguraci systému se ukázala cennou pro plánování jeho využití ve výuce – prakticky významné jsou zejména seznamy laboratorních zařízení v jednotlivých lokalitách a časové rozvrhy nabídek jejich sdílení.

Užitečnou vlastností generátoru konfigurací se ukazuje uchovávání starších verzí konfiguračních souborů. V případě, že je správcem některé lokality v databázi učiněna chybná změna a je vygenerována nesprávná konfigurace, může správce snadno a rychle obnovit činnost systému v některé z předchozích funkčních konfigurací výběrem některé z dřívějších konfigurací uchovávaných v archivu.

Vytvořením konfigurační aplikace se výrazně zvýšil komfort správců lokalit a omezilo množství konfiguračních chyb, které vznikaly jednak vlivem nekonzistence jednotlivých konfiguračních souborů a jednak vlivem překlepů při ruční editace konfiguračních souborů ve formátu XML. Nyní je konzistence garantována konfigurační aplikací, v jejímž GUI uživatel vybírá pouze z entit, které předtím sám (nebo některý z jeho kolegů definoval. Správcové lokalit, kteří chtějí do systému přidávat nové laboratorní prvky, již také nadále nemusí studovat syntaxe jednotlivých konfiguračních souborů a jejich návaznosti, ale pouze vyplní požadované informace o nově připojeném zařízení v přehledném webovém formuláři. Ukázka z GUI konfigurační aplikace je vidět na obr. 2.



Obrázek 2: Ukázka GUI konfigurační aplikace (definice parametrů laboratorního zařízení)

6 Kontrola aktivit uživatelů na laboratorních prvcích

Významným problémem při zajištění nepřetržité dostupnosti virtuální laboratoře se ukázala potřeba zabezpečit samotné laboratorní zařízení tak, aby je uživatelé nemohli svou konfigurací uvést do stavu, ve kterém s nimi nebudou moci další uživatelé pracovat. Jedná se zejména o různé formy vložení neznámého přístupového hesla, vymazání operačního systému nebo poškození předpřipravené výchozí konfigurace v paměti flash. Pokud by se zařízení do tohoto stavu dostalo, stalo by se nedostupným nejen pro pracující uživatele, ale i pro systém Virtlabu realizující automatické mazání konfigurací a obnovu konfigurace zařízení by musel pomocí zdlouhavé procedury provést správce systému manuálně.

Z těchto důvodů specifikovat množinu potenciálně nebezpečných příkazů a implementovat mechanismus, který jejich vložení zamezí. Z hlediska architektury systému se pro kontrolu zakázaných příkazů jevily vhodné dvě z jeho komponent: Java applet emulátoru terminálu, běžící ve WWW prohlížeči uživatelů systému a konzolový server, který běží v jednotlivých lokalitách a přeposílá příkazy z Java appletu přicházející přes TCP spojení jednotlivá laboratorní zařízení.

Přestože v prvních verzích byla kontrola zakázaných příkazů zabudována do appletu, z důvodu lepší odolnosti proti útokům jsme záhy kontrolu přesunuli do konzolového serveru, na němž je implementován jako samostatná komponenta. V implementaci jsme byli nuceni zohlednit také chování příkazového rozhraní zařízení Cisco (IOS), které v obou lokalitách pilotní instalace převládá. Jedná se zejména o možnost používání zkratek příkazů, doplňování textů příkazů tabulátorem a plné možnosti editace příkazového řádku vkládáním a mazáním znaků u kurzoru libovolně posunovatelného pomocí kurzorových kláves. Jelikož jsme nechtěli duplikovat funkci editace příkazového řádku ve vyhodnocovací komponentě, rozhodli jsme nechat skládání výsledného příkazového řádku na příkazovém interpreteru laboratorních zařízení a příkaz testovat na základě echa vysílaného ze zařízení do ovládacího terminálu, které v každém okamžiku odráží aktuální stav editovaného příkazu. Ten právě využívá vyhodnocovací komponenta v okamžiku, kdy uživatel stiskne znak `<ENTER>` k rozhodnutí, zda příslušný příkaz povolí nebo nikoli. V negativním případě se zaslání znaku `<ENTER>` do zařízení znemožní, čímž k vykonání zakázaného příkazu nedojde.

Množina zakázaných příkazů je uložena v databázi ve formě regulárních výrazů. V těch jsou zohledněny ne pouze plná znění zakázaných příkazů, ale i všechny jejich platné zkratky.

Při implementaci mechanismu vyhodnocování zakázaných příkazů bylo třeba rovněž zohlednit, že některé příkazy nelze zakázat jako celek, ale musíme dovolit jejich vložení s jistou konkrétní hodnotou parametru. Typicky se jedná o vložení pouze jediného konkrétního dohodnutého hesla. Situace je dále poněkud komplikována tím, že zatímco u příkazů vyhodnocovací systém nerozlišuje stejně jako Cisco IOS malá a velká písmena, u parametrů příkazů velikost písmen rozlišovat musí. Protože popis všech povolených variant regulárním výrazem by vedl k velmi rozsáhlým a nepřehledným výrazům se značnou pravděpodobností uživatelské chyby, rozhodli jsme se příkaz analyzovat ve dvou krocích. V prvním kroku vždy zjišťujeme, zda nejde o zakázaný příkaz a ve druhém kroku testujeme výjimky - povolenou hodnotu parametru zakázaného příkazu.

Protože virtuální laboratoř obsahuje různé typy síťových prvků, měli jsme v první fázi implementace představu, že seznam zakázaných příkazů bude přiřazen ke každému konkrétnímu fyzickému zařízení. V praxi se bohužel brzy ukázalo, že uživatel může k management

rozhraní síťových prvků přistoupit nejen přes jeho konzoli, ale také připojením pomocí SSH nebo Telnetu z jiného zařízení v laboratorní topologii. Proto jsme byli nuceni definovat jednu společnou množinu všech možných potenciálně nebezpečných příkazů, která je sjednocením zakázaných příkazů na všech zařízeních virtuální laboratoře. Tyto příkazy pak kontrolujeme a nedovolíme je uživateli zadat na žádném z laboratorních zařízení.

Abychom dokázali odhalit případné nebezpečné příkazy, které jsme mezi zakázané doposud nezařadili a které zapříčinily uvedení zařízení do nežádoucího stavu, ukládáme zachytáváme veškeré příkazy vkládané uživateli na laboratorních zařízeních do logovacího souboru. V případě problému jsme schopni příčinu dohledat a množinu zakázaných příkazů případně rozšířit. Pro vzdálené obnovení správné konfigurace jsme zakoupili jsme programově ovladatelný vypínač napájení síťových prvků, který nám dovolí libovolné zařízení vzdáleně restartovat a většinu poškození konfigurace manuálně vyřešit. V současné době zkoumáme možnosti vzdáleného uploadu operačního systému zařízení, pokud dojde i k jeho poškození.

V budoucích verzích virtuální laboratoře zvažujeme přenést odpovědnost za vkládání zakázaných příkazů přímo na jednotlivá laboratorní zařízení. Tuto možnost nabízí nové směrovače a přepínače firmy Cisco, které umožňují testovat vkládané příkazy pomocí ovladačů událostí v jazyce Tcl.

Pro případná zařízení jiných výrobců však bude nutné hledat jiná řešení.

7 Závěr

Pilotní konfigurace distribuované virtuální laboratoře vybudovaná společně VŠB-TU Ostrava a OPF-SLU Karviná je v současné době využívána studenty tamních akademií programu Cisco Networking Academy a současně jako povinná část distančního studia předmětů orientovaných na počítačové sítě v magisterském i bakalářském programu na FEI VŠB-TU. Laboratorní vybavení, které umožní oběma lokalitám specializaci na bezpečnostní, resp. VoIP technologie a sdílení nákladnějšího zařízení, bylo pořízeno s podporou projektu č. 213/2006 Fondu rozvoje sdružení Cesnet. Další rozšiřování možností implementace je postupně realizováno v rámci projektu č. 1212/2008 Fondu rozvoje vysokých škol.

Dosavadní zkušenosti z jejího provozu potvrdily předpokládané výhody plně distribuované architektury. Odhalily však také nutnost velmi dobré organizace zajištění pilotního provozu a náročnost jeho nepřetržitého a trvalého udrženi v takové kvalitě, na jakou jsou uživatelé dnešních komerčních webových aplikací zvyklí. Věříme, že některé ze zkušeností prezentovaných v tomto článku mohou být přínosné pro budování obdobných e-learningových systémů i pro realistické zhodnocení náročnosti i potřebných zdrojů pro jejich provoz.

Literatura

1. GRYGÁREK, Petr, MILATA, Martin. Piloting Environment of Distributed Virtual Networking Laboratory. In Virtual University. Bratislava (Slovensko) : [s.n.], 2007. s. 209-212. ISBN 9788089316090.
2. GRYGÁREK, Petr, MILATA, Martin, VAVŘÍČEK, Jan. The Fully Distributed Architecture of Virtual Network Laboratory. In ICETA. Stará Lesná (Slovensko) : [s.n.], 2007. ISBN 9788080860615.
3. GRYGÁREK, Petr, SEIDL, David, NĚMEC, Pavel. Zpřístupnění prvků laboratoře počítačových sítí pro praktickou výuku prostřednictvím Internetu. In Technologie pro e-vzdělávání. Praha : [s.n.], 2005. s. 43-52. ISBN 8001032744.
4. GRYGÁREK, Petr. Zkušenosti z nasazení virtuální laboratoře počítačových sítí a další směry jejího rozvoje. In Technologie pro e-vzdělávání. Praha : [s.n.], 2006. s. 58-68. ISBN 8001035123.
5. GRYGÁREK, Petr, SEIDL, David. Automatic WAN Topology Interconnection and it's Usage in CNAP Networking Laboratories. In ICETA, Stará Lesná (Slovensko) : [s.n.], 2007. ISBN 9788080860615.
6. Zabbix : The Ultimate Monitoring Solution [online]. 2008 [cit. 2008-04-06]. Dostupný z WWW: <http://www.zabbix.com/>.
7. Logování a debugging. VirtlabWiki [online]. 2008 [cit. 2008-04-06] Dostupný z WWW: <http://www.cs.vsb.cz/vl-wiki/index.php/Virtlab:LoggingDebugging>.
8. Monitor DVvirtlabu. 2008 [cit. 2008-04-06] Dostupný z WWW: <http://monitor.dvirtlab.net>.
9. PhpMyEdit : Instant MySQL Table Editor and PHP Code Generator [online]. c2006 [cit. 2008-04-06]. Dostupný z WWW: <http://www.phpmyedit.org/>.
10. Bugzilla.org [online]. c2008, Last modified February 1, 2008 [cit. 2008-04-06]. Dostupný z WWW: <http://www.bugzilla.org/>
11. Mantis [online]. 2008 [cit. 2008-04-06]. Dostupný z WWW: <http://www.mantisbt.org/>.
12. VirtIS : Jednoduchý systém pro evidenci jednotlivých úloh a nahlášených chyb pro distribuovaný VIRTLAB. 2008 [cit. 2008-04-06]. Dostupný z WWW: <http://virtis.viakis.net/>.
13. Eduroam.cz [online]. 2007 , Poslední úprava: 30.11.2007 11:58 [cit. 2008-04-06]. Dostupný z WWW: <http://www.eduroam.cz/>.
14. Shibboleth [online]. c2008 [cit. 2008-04-06]. Dostupný z WWW: <http://shibboleth.internet2.edu/>.
15. Generátor konfigurací : Jednoduchý systém pro generování konfiguračních souborů pro distribuovaný VIRTLAB. 2008 [cit. 2008-04-06]. Dostupný z WWW: <http://config.dvirtlab.net>.