

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

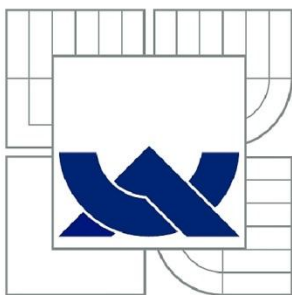
SPRÁVA STANIC PLANETLABU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

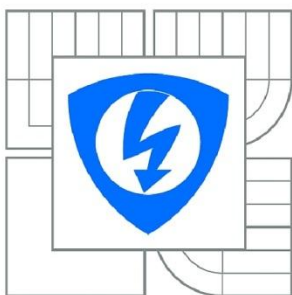
AUTOR PRÁCE
AUTHOR

MICHAL CHUPÁČ

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ**

**FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS**

SPRÁVA STANIC PLANETLABU

MANAGING PLANETLAB NODES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MICHAL CHUPÁČ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAKUB MÜLLER

BRNO 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Michal Chupáč

Ročník: 3

ID: 119458

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Správa stanic Planetlabu

POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je obsluha a správa stanic v síti Planetlab z jednoho místa. Nutností je zprovoznění vhodného nástroje pro upload souboru na vybrané uzly současně. Student by měl vhodně zobrazit funkčnost a stav používaných uzlů.

DOPORUČENÁ LITERATURA:

[1] PlanetLab Guide (online: <https://planet-lab.org/doc>)

[2] Navrátil J., PlanetLab - model budoucího Internetu. Zpravodaj ÚVT MU. ISSN 1212-0901, 2006, roč. XVI, č. 5, s. 1-5.

Termín zadání: 7.2.2011

Termín odevzdání: 2.6.2011

Vedoucí práce: Ing. Jakub Müller

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Bakalárska práca je zameraná na obsluhu a správu staníc PlanetLabu z jedného miesta. Úvodná kapitola obsahuje postup vytvorenia prístupu do siete a ďalšie dôležité informácie a rady. Nasledujúca časť sa venuje vzdialenému prístupu cez príkazový riadok pomocou programu CoDeploy. Tento program je využitý na paralelné vykonávanie príkazov a kopírovanie súborov na viacero uzlov súčasne. Súčasťou práce je návod na inštaláciu a konfiguráciu programov CoDeploy, Plush a Nebula ako aj popis ich funkcií a praktické využitie v prostredí siete PlanetLab. Časť s grafickým užívateľským rozhraním Nebula sa venuje vývojovému modulu XML aplikácií. Kompletne spracovaná je taktiež časť so softvérovou alternatívou PIMan určenou pre systémy Windows. Záver práce je venovaný monitorovaciemu systému CoMon, jeho funkciám, prvkom a praktickému nasadeniu v sieti.

ABSTRACT

Bachelor's thesis is focused on maintaining and managing PlanetLab nodes from one place. Introductory chapter contains process of network access creation and other important information and guidance. Following part involves remote access thru command prompt by program CoDeploy. This program is used for parallel command execution and file upload to multiple nodes. The thesis contains installation and configuration manual for applications CoDeploy, Plush and Nebula, description of their functions and practical usage in PlanetLab network environment. Section about graphic user interface Nebula deals with module for XML application development. Part with PIMan software alternative for Windows is also completely made out. Ending of the thesis is focused on CoMon monitoring system, its functions, elements and practical application in network.

KLÚČOVÉ SLOVÁ

Sieť PlanetLab, uzol, slice, SSH kľúčový pár, CoDeploy, nástroje Multiquery a Multicopy, web server Apache, Plush, grafické rozhranie Nebula, monitorovací systém CoMon, PIMan, modul na vývoj XML aplikácií, SWORD vyhľadávanie prostriedkov, CoVisualize, virtuálny server, formát csv, vzdialený prístup.

KEYWORDS

PlanetLab network, node, slice, SSH key pair, CoDeploy, Multiquery and Multicopy tools, web server Apache, Plush, graphic interface Nebula, CoMon monitoring system, PIMan, modul for XML application development, SWORD resource searching, CoVisualize, virtual server, csv format, remote access.

CHUPÁČ, M. *Správa stanic Planetlabu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 100 s. Vedúci bakalárskej práce Ing. Jakub Müller.

Prehlásenie

Prehlasujem, že svoju bakalársku prácu na tému Správa staníc PlanetLabu som vypracoval samostatne pod vedením vedúceho bakalárskej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Zb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia § 152 trestného zákona č. 140/1961 Zb.

V Brne dňa

.....

podpis autora

Pod'akovanie

Ďakujem vedúcemu bakalárskej práce Ing. Jakubovi Müllerovi za užitočnú metodickú pomoc a cenné rady pri spracovaní práce.

V Brne dňa

.....

podpis autora

OBSAH

ÚVOD	9
2 ZOZNÁMENIE SO SIEŤOU PLANETLAB	10
2.1 Čo je to PlanetLab	10
2.2 Terminológia	11
2.3 Vytvorenie prístupu k sieti PlanetLab	12
2.3.1 Vytvorenie SSH kľúča.....	12
2.3.2 Vytvorenie sliceu.....	13
2.4 Začíname pracovať zo sliceom.....	13
2.4.1 Riešenie problémov s prihlásením.....	14
2.4.2 Personalizácia sliceu.....	15
2.4.3 Upload aplikácie	15
2.4.4 Konfigurácia automatického spustenia služby	16
2.5 Udržiavanie sliceu a ďalšie užitočné informácie.....	17
2.5.1 Slice API.....	17
2.5.2 Obmedzenia	17
2.5.3 Etiketa a Netiketa.....	18
2.5.4 Alternatívne zdroje informácií.....	18
3 PRÍSTUP CEZ PRÍKAZOVÝ RIADOK	20
3.1 Úvod	20
3.2 Codeploy.....	20
3.2.1 Multiquery a Multicopy	21
3.2.2 Komponenty programu.....	21
3.2.3 Inštalácia a nastavenia	22
Inštalácia a nastavenie web serveru.....	24
3.2.4 Použitie programu CoDeploy	24
3.2.5 Použitie nástrojov Multiquery a Multicopy.....	25
3.3 Príklad využitia programu CoDeploy.....	26
3.3.1 Požiadavky	26
3.3.2 Zoznam serverov na upload.....	26
3.3.3 Príprava dát na web servery	27
3.3.4 Upload, spustenie a zastavenie skriptu	27
3.3.5 Zobrazenie získaných dát	28
3.4 Plush	29
3.4.1 Popis programu.....	29
3.4.2 Plush kontrolér.....	30
3.4.3 Požadované knižnice	30

3.4.4	Inštalácia a kompilácia	33
3.4.5	Nastavenie užívateľského prostredia	35
3.4.6	Spustenie.....	37
3.4.7	Využitie	37
4	GRAFICKÁ NADSTAVBA NEBULA	40
4.1	Inštalácia.....	40
4.2	Konfigurácia	41
4.3	Návod na použitie.....	42
4.3.1	Monitorovanie uzlov PlanetLabu	42
4.3.2	Zobrazenie uzlov sliceu	42
4.3.3	Správa sliceu.....	43
4.3.4	SSH prihlasovanie k uzlom	43
4.3.5	Vytvorenie a spustenie „HelloWorld“ aplikácie	44
4.3.6	Používanie SWORD vyhľadávania prostriedkov	44
4.3.7	Spúšťanie príkazov na zvolených uzloch	46
4.3.8	Určenie predka procesu	46
4.3.9	Využívanie softvérových a synchronizačných prvkov	47
4.3.10	Ďalšie funkcie aplikačného modulu	49
4.3.11	Praktické využitie programu.....	49
4.3.12	Proces pripojenia k uzlu	51
5	PLMAN: ALTERNATÍVA PRE WINDOWS	52
5.1	Popis programu.....	52
5.2	Inštalácia	53
5.3	Konfigurácia a spustenie	53
5.4	Funkcie programu.....	55
5.4.1	Výber uzlov na pripojenie	55
5.4.2	Spúšťanie príkazov	55
5.4.3	Pridanie uzlov k sliceu.....	55
5.4.4	Upload súborov	56
5.4.5	Download súborov.....	56
6	MONITORING	58
6.1	CoMon	58
6.1.1	Prehľad	58
6.1.2	Formát požiadaviek a výstupných dát	59
6.1.3	Vzorové požiadavky	59
6.1.4	Prístup k dátam	61
6.1.5	Popis prvkov html rozhrania.....	62
6.2	CoVisualize	66
6.2.1	Vizualizácie	66

6.3	Využitie monitoringu.....	69
6.3.1	Slice princeton_comon	69
6.3.2	Výskumná sieť CESNET.....	70
6.3.3	Rôzne problémy v sieti.....	71
6.3.4	Vlastný výber a spracovanie dát.....	72
7	ZÁVER	73
	POUŽITÁ LITERATÚRA	75
	ZOZNAM SKRATIEK A JEDNOTIEK	76
	ZOZNAM PRÍKAZOV	78
	ZOZNAM OBRÁZKOV	78
	ZOZNAM PRÍLOH	79
8	PRVÁ PRÍLOHA	80
9	DRUHÁ PRÍLOHA	84
10	TRETIA PRÍLOHA	91
11	ŠTVRTÁ PRÍLOHA	94

ÚVOD

Úlohou bakalárskej práce je vyžitie vhodného programového vybavenia na obsluhu a právu staníc PlanetLabu z jedného miesta.

Úvodná časť približuje štruktúru siete, jej ciele a venuje sa vytvoreniu prístupu k jej uzlom. V závere sú uvedené zásady pre správu pridelených prostriedkov ako aj ďalšie užitočné informácie potrebné pri riešení prípadných problémov.

Nasledujúca časť je zameraná na vzdialený prístup cez príkazový riadok a obsahuje návody na inštaláciu a konfiguráciu nástrojov CoDeploy a Plush. Popísané sú aj funkcie obidvoch programov. Časť týkajúca sa programu CoDeploy obsahuje aj jeho praktické využitie na paralelné spúšťanie príkazov a upload súborov na viacero uzlov súčasne.

Štvrtá kapitola je venovaná nadstavbovému grafickému rozhraniu programu Plush s názvom Nebula. Okrem spracovania kompletného návodu ako u predošlých programov je táto časť venovaná hlavne vývojovému prostrediu XML aplikácií a jeho praktickému využitiu.

Obsahom ďalšej časti je spracovanie podkladov softvérovej alternatívy pre systémy Windows s názvom PIMan. Uvedená inštalácia a nastavenie programu obsahuje aj pokyny pre linuxové distribúcie.

Záverečná časť obsahuje popis prostredia a funkcií monitorovacieho systému CoMon a jeho grafickej nadstavby CoVisualize. Na konci kapitoly je uvedených niekoľko príkladov využitia tohto systému na monitorovanie sieťových uzlov a odhalenie príčiny vzniknutých problémov alebo podozrivého chovania. Taktiež sú uvedené spôsoby prístupu a získavania dát na ďalšie spracovanie.

V závere sú zhrnuté dosiahnuté ciele a podrobnejší popis obsahu spracovaných návodov a výsledkov praktických častí.

Prílohy obsahujú potrebné skripty a zobrazenia prostredí použitých služieb a aplikácií.

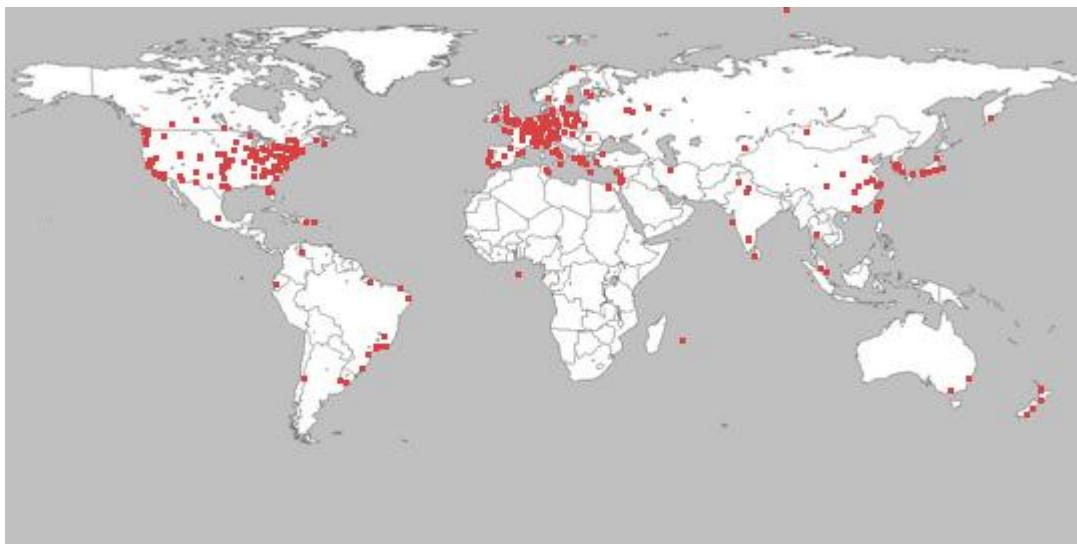
2 ZOZNÁMENIE SO SIEŤOU PLANETLAB

2.1 Čo je to PlanetLab

PlanetLab je celosvetová výskumná sieť podporujúca vývoj nových sieťových služieb. Od roku 2003 sa do tohto vývoja zapojilo viac ako tisíc výskumných pracovníkov vrcholových akademických organizácií a priemyselných výskumných laboratórií. Organizácie, ktoré sa dlhodobo angažovali vo vývoji hardvéru a softvéru siete PlanetLab vytvorili konzorcium poskytujúce podporu pre ostatných užívateľov a zodpovedné za ďalší vývoj výskumnej siete. Táto spolupráca viedla k vytvoreniu softvérového balíčku MyPLC slúžiaceho na jednoduché vytvorenie vlastných PlanetLab severov v členských výskumných organizáciách. To umožnilo ďalej rozvíjať bežné balíčky služieb uľahčujúcich vývoj nadstavbových služieb a aplikácií nielen pre koncových užívateľov ale aj pre zdokonalenie samotnej siete PlanetLab.

Hlavným cieľom PlanetLabu je okrem podpory výskumu a testovania v reálnych podmienkach siete aj sprístupnenie vyvíjaných služieb koncovým užívateľom pre uľahčenie tvorby vlastných aplikácií. Podpora obidvoch úloh je považovaná za kritickú pre ďalší vývoj a úspech systému.

V súčasnosti sieť PlanetLabu pozostáva z 1065 uzlov v 561 organizáciách, ktoré pracujú na viac ako 600 aktívnych projektoch.



Obr. 2.1: Celosvetové rozšírenie organizácií PlanetLabu.

2.2 Terminológia

Nasleduje zoznam bežných termínov použitých v tomto dokumente a vysvetlenie ich významu alebo úlohy v sieti PlanetLab.

Organizácia (Site): fyzické miesto kde sa uzly PlanetLabu nachádzajú (napr. CESNET - Czech Education and Research Network). Zjednodušená verzia názvu miesta je prefixom názvov všetkých sliceov.

Uzol (Node): vzdialený server, na ktorom sú spustené komponenty služby PlanetLab.

Slice: sada pridelených zdrojov distribuovaných v sieti PlanetLab. Pre väčšinu užívateľov je slice konzolovým prístupom k serverom PlanetLabu. Vedúci projektu je zodpovedný za vytvorenie sliceov a ich pridelenie k užívateľom. Po pridelení sliceu si k nemu môže užívateľ pridať uzly. Následne po pridaní uzlov k sliceu je na každom pridelenom uzle vytvorený príslušný virtuálny server. Slice má určitú dobu platnosti, ktorá musí byť pravidelne obnovovaná.

Sliver: pridelené zdroje na jednom uzle PlanetLabu.

Virtuálny server (VS): slivery sú implementované ako linuxové virtuálne servery. VS realizuje oddelenie menného priestoru a výkonu medzi slivermi na jednom stroji. Sieťovú virtualizáciu sliverov umožňuje modul prístupu k virtuálnej sieti (VNET). Príležitostne môže dôjsť k zámene termínov slice, sliver, VS pričom ale nesmieme zabudnúť na rozdiely v ich architektúre.

Vedúci projektu (Principal Investigator - PI): vedúci projektu jednotlivých organizácií zodpovedný za správu sliceov a kontrolu ich užívateľov. Taktiež je právne zodpovedný za obsah sliceov, ktoré užívatelia vytvoria. Väčšina organizácií má len jedného vedúceho projektu.

Správca (Technical Contact - TC): každá organizácia je povinná mať aspoň jedného správcu, ktorý je zodpovedný za inštaláciu, správu a monitorovanie sieťových uzlov organizácie.

Užívateľ (User): užívateľom je každý kto využíva, vyvíja a inštaluje aplikácie na uzly PlanetLabu. Užívateľom môže byť aj vedúci projektu.

PlanetLab Central API (Advanced Programmatic Interface - PLCAPI): programové rozhranie, ktoré využíva web stránka PlanetLabu, uzly, automatické skripty a užívatelia na prístup a úpravu údajov PlanetLab databázy.

2.3 Vytvorenie prístupu k sieti PlanetLab

Aby Ste mohli začať pracovať, musíte si najskôr vytvoriť vlastný PlanetLab účet na tejto registračnej *stránke*¹. Prečítajte si užívateľské *podmienky*². Ak súhlasíte s ich dodržiavaním môžete si vytvoriť účet.

Ak Vaša organizácia nie je v zozname na registračnej stránke kontaktujte svojho správcu a uistite sa, že jej bola povolená účasť v projekte PlanetLab. Pokiaľ je všetko v poriadku vyplňte požadované informácie a potvrdte registráciu (e-mailová adresa je zároveň Vaším užívateľským menom v sieti PlanetLab). Požiadavka na registráciu bude zaslaná na potvrdenie vedúcemu projektu vo Vašej organizácii. Po potvrdení obdržíte e-mail. V tomto okamžiku môžete pokračovať podľa ďalšieho postupu.

Ak niekedy zabudnete svoje heslo, môžete požiadať o jeho zaslanie na e-mail pomocou *formuláru*³ na reset hesla.

2.3.1 Vytvorenie SSH kľúča

Pred tým ako budete môcť pristupovať k serverom PlanetLabu, vrátane tých vo Vašej organizácii, musíte si vytvoriť dvojicu SSH kľúčov potrebnú pre autentifikáciu.

Vzdialený prístup k serverom PlanetLabu je obmedzený na SSH prihlásenie pomocou RSA autentifikácie, ktorá je založená na šifrovaní verejného kľúča. Kódovanie a dekódovanie je realizované odlišnými kľúčmi pričom nie je možné odvodiť dekódovací kľúč z kľúča použitého na kódovanie. Na vytvorenie páru kľúčov použite program „ssh-keygen“ na zabezpečenom UNIX systéme.

Príkaz 2.1: Vytvorenie SSH kľúča.

```
$ ssh-keygen -t rsa -f ~/.ssh/id_rsa
```

Po zadaní uvedeného príkazu budete požiadaný o zadanie passphrase, na základe ktorej sa vygenerujú súbory s kľúčmi: súbor so súkromným kľúčom „id_rsa“ a súbor s verejným kľúčom „id_rsa.pub“. Kľúč musí byť vo formáte OpenSSH. Ak prvý riadok súboru s príponou „.pub“ nezačína reťazcom znakov `ssh-rsa AAAAB3Nza...` musíte kľúč prekonvertovať. Prečítajte si dokumentáciu dodávanú s Vašou verziou programu „ssh-keygen“. Dobré heslo by malo byť dlhé 20-30 znakov a nemalo by byť ľahko odhadnuteľné. Súbor so súkromným kľúčom „id_rsa“ si uschovajte na bezpečnom mieste,

¹ <https://www.planet-lab.org/db/persons/register.php>

² <http://www.planet-lab.org/aup/>

³ https://www.planet-lab.org/db/persons/reset_password.php

napríklad na flash disku. Súbor s verejným kľúčom „id_rsa.pub“ nahrajte na stránku PlanetLabu do *sekcie*¹ správy kľúčov.

Dávajte si pozor na súbor s Vaším súkromným kľúčom! Ak dôjde k jeho odcudzeniu a Váš účet bude zneužitý na nelegálne aktivity Vaša organizácia a správca budú za to právne zodpovední. Preto ak svoj kľúč stratíte alebo máte pocit, že Vám ho ukradli alebo inak skompromitovali, vygenerujte si nový pár kľúčov s použitím nového hesla, nahrajte súbor s verejným kľúčom na web stránku a okamžite upovedomte technickú *podporu*² PlanetLabu.

2.3.2 Vytvorenie sliceu

Po nahraní Vášho verejného kľúču na web stránku požiadajte vedúceho projektu o priradenie Vášho účtu k existujúcemu sliceu alebo o vytvorenie nového. Názov sliceu bude začínať zjednodušeným názvom Vašej organizácie, napr. „cesnet_s2“. Po vytvorení a pridelení k sliceu si môžete k nemu prideliť servery pomocou *formuláru*³ na správu uzlov. Môže trvať aj niekoľko hodín, kým dôjde k vytvoreniu Vášho sliceu na všetkých serveroch a k rozšíreniu Vášho verejného kľúča. Ak bol už Váš slice na servery vytvorený môžete sa k nemu prihlásiť podľa inštrukcií v časti 2.4. Samozrejme k serverom, ktoré boli počas vytvárania sliceu vypnuté, sa nebudete môcť pripojiť. Bude trvať ešte niekoľko minút kým sa server znovu spustí a vytvorí sa na ňom Váš slice.

2.4 Začíname pracovať zo sliceom

Vytvorený slice obsahuje minimálnu inštaláciu linuxovej *distribúcie*⁴ Fedora Core 8. Pre prihlásenie na server pomocou SSH uveďte názov sliceu ako užívateľské meno (napr. „cesnet_s2“), cestu k súboru s Vaším privátnym kľúčom (napr. „~/.ssh/id_rsa“) a názov severu, na ktorý sa chcete prihlásiť (napr. „planetlab2.cesnet.cz“).

Príkaz 2.2: Prihlásenie k vzdialenému uzlu.

```
$ ssh -l cesnet_s2 -i ~/.ssh/id_rsa planetlab2.cesnet.cz
```

Budete prihlásený ako bežný užívateľ s rovnakým menom ako je názov Vášho sliceu. Pretože slice je vytvorený v bezpečnom, izolovanom prostredí, je Vám povolený obmedzený prístup roota a môžete sa pomocou príkazu „su“ prepnúť do koreňového adresára.

¹ <https://www.planet-lab.org/db/persons/index.php>

² support@planet-lab.org

³ <https://www.planet-lab.org/db/nodes/index.php>

⁴ <http://www.redhat.com/Fedora/>

Takisto môžete pridávať nových užívateľov, spravovať služby, inštalovať nové balíčky, pripojiť určité adresáre a vykonávať ďalšie úkony, ktoré sú bežne rootovi povolené.

Avšak, väčšina administratívnych a privilegovaných operácií, ako napr. konfigurácia hardvéru a siete, nie je povolená ani rootovi. Niektoré rozhrania, ako napr. sokety raw paketov, povoľujú prístup k virtuálnemu hardvéru z dôvodu kompatibility, sú ale prísne kontrolované. Bližšie informácie ohľadom virtualizácie sieťového prístupu pre Váš slice nájdete v *príručke*¹ k modulu VNET. Ak chcete vykonávať iné privilegované operácie, ako napr. pristupovať k súborovým systémom ostatných sliceov a otvárať raw sokety, použite na to určenú *aplikáciu*².

Aj keď je väčšina administratívnych operácií sliceom zakázaná a majú obmedzené využitie zdrojov, môže sa stať, že dôjde k odstaveniu uzlu, siete organizácie alebo časti internetu zahľtením prenosového pásma. Takéto správanie sliceu má za následok automatické spustenie detekčného systému (IDS) ale taktiež môže prilákať hackerov. V prípade rozsiahlejšieho problému môžu zakročiť aj právne zodpovedné úrady alebo osoby, **preto je obzvlášť dôležité aby Ste si uvedomili aká dôležitá je Vaša zodpovednosť udržiavania sliceu pod kontrolou a v bezpečí. Odporúčame preventívne kontrolovať aktivitu Vášho sliceu a častú zmenu SSH kľúčov.**

2.4.1 Riešenie problémov s prihlásením

Z rôznych dôvodov ako je napríklad zlyhanie hardvéru dochádza k vyradeniu serveru. V prípade, že server je dočasne alebo nastalo mimo prevádzku, nebudete sa môcť k nemu pripojiť. Majte na pamäti, že systém na ktoromkoľvek servery môže byť preinštalovaný alebo môže dôjsť k výmene serveru. **Preto by Ste nemali servery využívať na permanentné uloženie dát!**

Ak sa nemôžete prihlásiť k uzlu, ktorý bol pridelený k Vášmu sliceu, skontrolujte či Ste ako prihlasovacie meno použili názov sliceu. Pridajte parameter „-v“ za príkaz „ssh“ do príkazového riadku pre zobrazenie podrobností prihlasovacieho procesu.

Príkaz 2.3: Riešenie problémov s prihlásením.

```
$ ssh -v -l cesnet_s2 -i ~/.ssh/id_rsa planetlab2.cesnet.cz
```

Ďalším bežným problémom sú nesúhlasné host kľúče. SSH kľúče PlanetLab serverov sa občas menia. Ak sa v zozname hostiteľských kľúčov („~/.ssh/known_hosts“) na Vašom počítači nachádza starý záznam je potreba záznam alebo celý súbor zmazať.

¹ <https://www.planet-lab.org/doc/vnet/>

² <https://www.planet-lab.org/software/>

Ak sa stále nemôžete prihlásiť, použite aspoň jeden z prístupných *nástrojov*¹ na zistenie statusu serverov predtým ako sa obrátite na technickú *podporu*² PlanetLabu. Server môže byť dočasne mimo prevádzku. Pri kontakte s technickou podporou ohľadom neúspešného prihlásenia uveďte názov Vášho sliceu, názov uzlu a rozšírený výpis ssh prihlasovania.

2.4.2 Personalizácia sliceu

Z dôvodu úspory miesta je väčšina súborov a systémových knižníc tvorená len priamymi odkazmi na ich kópie. Z toho vyplýva, že tieto súbory nemôžete meniť. Je ich však možné zmazať a nahradiť, čo je bežný aktualizčný postup inštalačných manažérov ako je napr. „rpm“. Priame odkazy sa nezapočítavajú do celkového využitia diskového priestoru, na rozdiel od ktoréhokoľvek súboru, ktorý nahradíte, nainštalujete alebo iným spôsobom vytvoríte.

Na inštaláciu prídavných štandardných knižníc na Váš slice, použite nástroj yum. Napríklad pre nainštalovanie nástroja emacs zadajte príkaz `yum install emacs` a pre aktualizáciu všetkých nainštalovaných knižníc príkaz `yum update`.

Bližšie informácie o možnostiach využitia nástroja yum nájdete v jeho manuáli. Ak využívate tento nástroj na hromadnú aktualizáciu všetkých uzlov sliceu zvážte možnosť úpravy základného konfiguračného súboru „`/etc/yum.conf`“ nahradením záznamov „`FedoraCore2Base`“ a „`FedoraCore2Updates`“ alternatívnymi verejnými odkazmi pre zníženie zaťaženia PlanetLab boot serveru.

2.4.3 Upload aplikácie

Po vytvorení sliceu a nainštalovaní potrebných nástrojov môžete pristúpiť k uploadu Vašej aplikácie. Aby Ste sa uistili, že Vašu aplikáciu bude možné na PlanetLab servery spustiť, pred uploadom na server ju skompilujte a otestujte na počítači z nainštalovanou distribúciou Fedora Core 2. Najjednoduchším spôsobom uploadu aplikácie na konkrétny server je použitie nástroja scp.

Príkaz 2.4: Upload súborov pomocou nástroja scp.

```
$ scp -i ~/.ssh/id_rsa -r test cesnet_s2@planetlab2.cesnet.cz:
```

Uvedený príkaz slúži na skopírovanie lokálneho adresáru „test“ do domovského adresáru PlanetLab serveru s názvom „planetlab2.cesnet.cz“, ktorý je súčasťou sliceu „cesnet_s2“.

¹ <http://www.planet-lab.org/status/>

² support@planet-lab.org

Nástroj rsync môžete použiť na skopírovanie nových alebo zmenených súborov.

Príkaz 2.5: Aktualizácia súborov nástrojom rsync.

```
$ rsync -a -e "ssh -l cesnet_s2 -i ~/.ssh/id_rsa" test  
planetlab2.cesnet.cz:
```

Užívatelia PlanetLabu vytvorili mnoho ďalších nástrojov určených na upload a spustenie nových aplikácií. Zoznam dostupných nástrojov nájdete na stránke s užívateľskými *nástrojmi*¹.

2.4.4 Konfigurácia automatického spustenia služby

Pri plánovanej údržbe sú servery pravidelne reštartované. Pokiaľ je Vaša aplikácia službou bežiacou na pozadí je potreba nastaviť jej automatické spúšťanie po spustení serveru a korektné ukončenie pred jeho vypnutím. K tomu je potrebné vytvoriť inicializačný skript v „/etc/rc.d/init.d“. Môžete použiť akýkoľvek existujúci skript ako vzor aby sa dalo ďalej so skriptom pracovať pomocou nástroja chkconfig. Spustenie a ukončenie aplikácie v konkrétnom poradí medzi ostatnými službami môžete nastaviť úpravou riadku začínajúceho „# chkconfig:“, ktorý je umiestnený v počiatočnej časti skriptu. Prvé číslo vyjadruje úroveň spustenia skriptu, druhé číslo je relatívnym poradím spustenia a tretie poradím pri ukončovaní. Váš slice je vždy spustený v úrovni 3 a ukončený v úrovni 0. Zobrazte si obsah súborov „/etc/rc3.d“ a „/etc/rc0.d“ pre určenie poradia služieb.

Vytvorený skript môžete povoliť pomocou nástroja chkconfig.

Príkaz 2.6: Povolenie skriptu pomocou nástroja chkconfig.

```
$ chkconfig --add test  
$ chkconfig test on
```

Po nainštalovaní je možné vytvorený skript spustiť a ukončiť pomocou nástroja service.

Príkaz 2.7: Spustenie/ukončenie skriptu nástrojom service.

```
$ service test start  
$ service test stop
```

Týmto spôsobom bude systém spúšťať Váš skript pri štarte a ukončení.

Ak Vám nezáleží na vytvorení vlastného procesu spúšťania a ukončenia postačí, keď editujete skript „/etc/rc.d/rc.local“ pridaním príslušného príkazu na štart a ukončenie Vašej aplikácie. Skript „rc.local“ sa spúšťa pri štarte ako posledný.

¹ <https://www.planet-lab.org/tools/>

2.5 Udržiavanie sliceu a ďalšie užitočné informácie

2.5.1 Slice API

Každodenne dochádza k vytváraniu nových a k výmene, prípadne odstaveniu, starých serverov PlanetLabu. Na uľahčenie rutiny pridávania a odoberania uzlov zo sliceu odporúčame použiť namiesto web stránky PlanetLabu rozhranie *API*¹. Stránka PlanetLabu sama používa toto rozhranie takže pomocou neho budete môcť vykonávať tie isté operácie aké sú Vám umožnené vo web rozhraní.

2.5.2 Obmedzenia

Slice je vytvorený na požiadanie a má platnosť dva mesiace. Po uplynutí doby platnosti dôjde k zmazaniu obsahu sliceu a k jeho odstráneniu zo všetkých pridelených uzlov. Doba platnosti sliceu sa dá predĺžiť pomocou *formuláru*² na obnovu sliceu. Počet predĺžení platnosti nie je limitovaný ale maximálna doba platnosti nemôže byť dlhšia ako dva mesiace. Ak už nebudete slice využívať môžete ho nechať expirovať, ale lepšie je ho zmazať a tým uvoľniť jeho prostriedky pre ostatných užívateľov. Slice má tiež limitovaný priestor na disku, využitie pamäte, popisovačov súborov a šírku prenosového pásma.

Expiračná doba Vášho sliceu môže byť upravená administrátorom z rôznych dôvodov. Ak je napr. popis Vášho sliceu nedostatočný alebo obsahuje nefunkčný odkaz na web stránku Vášho projektu, admin môže nastaviť expiráciu na pár dní aby Vás upozornil na nutnosť úpravy uvedených informácií. **Ak do uplynutia doby platnosti popis neupravíte Váš slice bude zmazaný!**

Vytváranie sliceov a ich obnova môže byť tiež dočasne zakázaná z dôvodu dlhodobejšej nedostupnosti uzlov alebo ich odstavenia. Ak k tomu dôjde kontaktujte svojho PI alebo správcu. Ak Ste si istý, že uzly Vášho sliceu sú už aktívne a dostupné kontaktujte *podporu*³ PlanetLabu. Ak nie, navštívte *stránku*⁴ s podrobnými informáciami o Vašej organizácii, kde sa dozviete bližšie informácie o stave uzlov a taktiež nájdete kontakt na PI a správcu.

¹ http://www.planet-lab.org/doc/plc_api/

² <http://www.planet-lab.org/db/slices/index.php>

³ support@planet-lab.org

⁴ <https://www.planet-lab.org/db/sites/index.php>

2.5.3 Etiketa a Netiketa

Prostredia jednotlivých sliceov sú navzájom oddelené čo zabraňuje monitorovaniu aktivity iných sliceov pomocou základných nástrojov ako je top alebo ps. Aj napriek obmedzeniu jednotlivých zdrojov sliceu môže však stále dôjsť k narušeniu činnosti ostatných sliceov zahltením niektorej s poskytovaných služieb. Uvediem niekoľko preventívnych opatrení ako sa vyhnúť nežiaducim problémom:

- Vytvorte stránku, ktorá bude obsahovať ciele Vášho sliceu spolu s kontaktnými informáciami na vedúcich pracovníkov. Odkaz na túto stránku vložte do popisu sliceu.
- Ak je Vami vytvorená aplikácia úspešná naplánujte si dopredu spôsob jej údržby a podpory v prípade rozšírenia.
- Dôkladne vylad'ite svoj kód, obzvlášť jeho núdzový režim, pred jeho nasadením. Nepretržite kontrolujte jeho chod. Použite nástroj *CoMon*¹ na monitorovanie stavu uzlov, na ktorých je vytvorený Váš slice a zaznamenávajúte anomálie.
- Nevykonávajte systematické skenovanie portov alebo iné aktivity, ktoré by mohli byť považované za útok alebo inú nelegálnu činnosť. Takéto aktivity sú považované za porušenie užívateľských *podmienok*² a Ste za ne právne zodpovedný.
- Veľké množstvo strojov po celom svete podieľajúcich sa na stovkách projektov vytvára veľký rozruch. Uzly sú reštartované organizáciami a technikmi PlanetLabu z rozličných dôvodov. Niekedy si softvérová, hardvérová chyba alebo upgrade operačného systému vyžaduje kompletnú reinstaláciu aby mohol server ďalej fungovať. **Z toho dôvodu nezabúdajte svoje dáta pravidelne zálohovať mimo sieť. Servery PlanetLabu nie sú zálohované!**

2.5.4 Alternatívne zdroje informácií

Komunita PlanetLabu pozostáva z viac ako 1000 užívateľov. V prípade akýchkoľvek otázok ohľadom kompilácie, uploadu a spustenia aplikácie sa môžete obrátiť na hocikoho zo *zoznamu*³ užívateľských e-mailov. Určite Vám rád pomôže, prípadne Vám poradí na koho sa môžete obrátiť.

¹ <http://comon.cs.princeton.edu>

² <http://www.planet-lab.org/aup/>

³ <http://lists.planet-lab.org/mailman/listinfo/users/>

Ak máte technické problémy s uzlom, sliceom, účtom alebo web stránkou kontaktujte podporu¹ PlanetLabu. Ak si zriadíte službu, ktorá bude dlhodobo spustená, vyzveme Vás a budeme časom požadovať aby Ste sa zapojili do asistenčného systému na zodpovedanie otázok a riešenie pripomienok a návrhov súvisiacich s Vašou službou.

Súčasný prostredie platformy PlanetLabu je Linux, takže zručnosť pri práci s programovým prostredím Linuxu a UNIXu je základom. Pomoc užívateľom s otázkami súvisiacimi s týmito operačnými systémami je nad rámec možností podpory PlanetLabu. Ak máte podobnú otázku pokúste sa ju vyriešiť pomocou služby Google² alebo cez archívne záznamy³ užívateľských e-mailov. Až následne v prípade nevyhnutnosti kontaktujte podporu PlanetLabu.

¹ support@planet-lab.org

² <http://www.google.com>

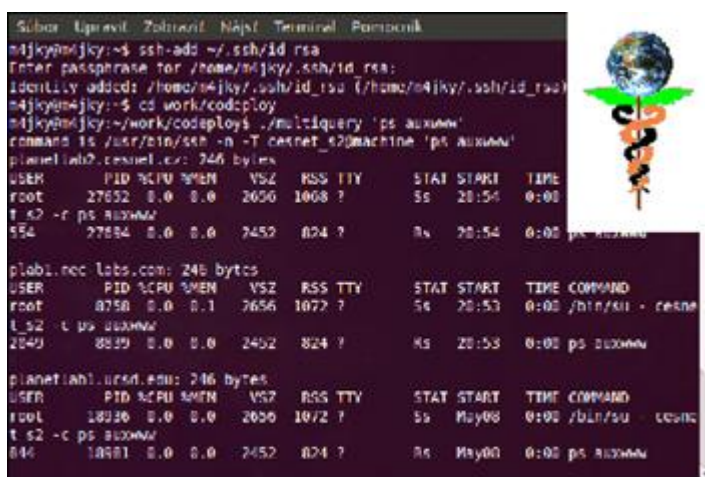
³ <http://lists.planet-lab.org/pipermail/users/>

3 PRÍSTUP CEZ PRÍKAZOVÝ RIADOK

3.1 Úvod

V tejto časti sa budem zaoberať vzdialeným prístupom a prácou s viacerými servermi cez príkazový riadok. Na tento účel vyberiem niekoľko aplikácií a uvediem možnosti ich využitia v sieti PlanetLab. Popíšem postup ich inštalácie a konfigurácie a uvediem príklad ich praktického využitia.

3.2 Codeploy



```
mikey@mikey:~$ ssh-add ~/.ssh/id_rsa
Enter passphrase for ~/.ssh/id_rsa:
Identity added: ~/.ssh/id_rsa (/home/mikey/.ssh/id_rsa)
mikey@mikey:~$ cd work/codeploy
mikey@mikey:~/work/codeploy$ ./multiquery 'ps auxwww'
command is /usr/bin/ssh -n -T cecnet_02@machine 'ps auxwww'
planetlab2.cesnet.cz: 246 bytes
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      27632  0.0  0.0   2056  1008 ?        Ss   20:54   0:00
t_k2 -r ps auxwww
954      27694  0.0  0.0   2452    824 ?        Rs   20:54   0:00 ps auxwww

planetlab1.nec.labs.com: 246 bytes
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      8758  0.0  0.1   2656  1072 ?        Ss   20:53   0:00 /bin/su - cecnet
t_s2 -c ps auxwww
2049      8839  0.0  0.0   2452    824 ?        Rs   20:53   0:00 ps auxwww

planetlab1.ured.edu: 246 bytes
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      18236  0.0  0.0   2056  1072 ?        Ss   May08   0:00 /bin/su - cecnet
t_s2 -c ps auxwww
844      18581  0.0  0.0   2452    824 ?        Rs   May08   0:00 ps auxwww
```

Obr. 3.1: Terminálové prostredie programu.

Program CoDeploy poskytuje prostriedky na efektívnu distribúciu ľubovoľného obsahu z jedného zdroja k viacerým príjemcom. V sieti PlanetLab to prakticky znamená možnosť hromadného odoslania dát na stovky uzlov bez spôsobenia enormného zaťaženia na strane odosielateľa. CoDeploy na efektívny prenos dát využíva nasledovné techniky:

- Využitie uchovávaní dát v HTTP cache pri prenose cez distribučnú sieť CoDeeN.
- Umožnenie efektívneho prenosu veľkých súborov ich delením na menšie časti.
- Vyhľadanie vhodných distribučných serverov pomocou monitorovacieho systému „peer review“.

Dôvodom výberu uvedeného nástroja je jeho jednoduchosť, rýchla inštalácia a nenáročnosť na pracovné prostredie. Na realizáciu prenosu postačuje prítomnosť dvoch aplikácií na počítači zahajujúcom prenos. Ostatné prostriedky sú k dispozícii v sieti PlanetLab. Proces prenosu funguje nasledovne:

1. Pomocou Web serveru poskytneme adresár obsahujúci dáta na prenos.
2. Zadáme zoznam cieľových uzlov.
3. CoDeploy vygeneruje informácie o verzii zasielaných dát pomocou kontrolného súčtu a rozošle ho zadaným uzlom.

4. Následne každý cieľový uzol porovná svoju verziu súboru s verziou dostupnou na zdrojovom servery a v prípade nezhody zaháji prenos.
5. Codeploy kontaktuje miestnu službu spustenú na každom uzle. Táto služba nasmeruje program CoDeploy k najbližšiemu funkčnému distribučnému serveru siete CoDeeN, ktorý je následne využitý na sťahovanie dát.
6. V priebehu sťahovania je pôvodný súbor rozdelený na množstvo malých častí a duplikovaný na všetky dostupné distribučné servery. Tieto fragmenty sú pri následnom vyžiadaní znovu zložené.

3.2.1 Multiquery a Multicopy

Súčasťou inštalačného balíčka programu CoDeploy sú aj ďalšie dva nezávislé nástroje Multiquery a Multicopy. Nástroj Multiquery slúži na paralelné vykonávanie jednoduchých shell operácií na viacerých vzdialených uzloch pomocou ssh. Multicopy na prenos súborov z jedného zdroja k viacerým cieľom alebo opačne.

3.2.2 Komponenty programu

Program CoDeploy pozostáva z niekoľkých komponentov podieľajúcich sa na procese prenosu dát. Uvediem ich stručný popis:

Generátor skriptu – program CoDeploy pracuje pomocou dynamicky generovaných skriptov. Po spustení prenosu adresáru, generátor skriptu vytvorí záznamy o jednotlivých podadresároch, súboroch a ich kontrolných súčtoch (md5sum). Získané informácie sú použité na vytvorenie perl skriptu, ktorý je možné následne spustiť na jednotlivých uzloch PlanetLabu.

Paralelné spustenie/kopírovanie – vytvorený skript je na cieľové uzly následne skopírovaný a spustený pomocou nástrojov Multicopy a Multiquery. Skript vytvorí potrebné adresáre a porovná kontrolné súčty existujúcich súborov so zdrojovými súbormi. Každý neaktuálny súbor je následne označený na stiahnutie cez sieť CoDeeN.

Služba peer-review – z dôvodu využitia distribučných serverov je potrebný mechanizmus na monitorovanie ich stavu. Na tieto účely je na jednotlivých serveroch spustená monitorovacia služba, ktorá slúži na vzájomnú kontrolu ich stavu. Každý server tak dostáva správy hodnotiace jeho stav a rozosiela správy o svojej reputácii, ktoré sú následne využívané na priradenie váhy hodnotiacim správam.

CoDeeN lokalizátor – keďže nie každý PlanetLab server je zaradený do distribučnej siete, CoDeploy potrebuje mechanizmus na vyhľadanie najbližšieho funkčného CoDeeN serveru. Lokalizačná služba zbiera stavové

správy od známych distribučných serverov a získané údaje následne kombinuje s údajmi o polohe serverov. Výsledkom je informácia pre program CoDeploy obsahujúca najvhodnejší server na použitie.

Podpora veľkých súborov – každý server poskytuje službu, ktorá prijíma požiadavky na sťahovanie a následne vytvára sadu požiadaviek o jednotlivé časti súboru. Tieto požiadavky sú následne rozoslané na distribučné servery, ktoré v prípade potreby nezávisle vyžadujú časti súboru zo zdroja. Tieto malé časti sú následne skompletizované serverom, ktorý požiadavky rozoslal a späťne zaslané klientovi.

3.2.3 Inštalácia a nastavenia

Ďalej uvedený postup bol vypracovaný pre Ubuntu 10.10 x64. V prípade, že využivate inú distribúciu použite ako oporu pôvodný *návod*¹ uverejnený na web stránke programu CoDeploy.

Pri inštalácii a konfigurácii postupujte nasledovne:

1. Stiahnite si *balíček*² so zdrojovými súborami programu.
2. Súbor rozbaľte príkazom `tar xzf codeploy.tar.gz`. Tým vytvoríme adresár s názvom codeploy.
3. Zadaním príkazu `make` v rozbalenom adresári vytvorte potrebné súbory.
4. Vytvorte zoznam, ktorý bude obsahovať názvy Vami zvolených uzlov, jeden názov na riadok. V prípade, že potrebujete vybrať väčší počet uzlov na základe konkrétnych požiadaviek môžete využiť službu CoMon. Výber potrebných uzlov môžete realizovať pridaním zvolených premenných do cgi skriptu programu. Bližšie informácie ohľadom použitia služby CoMon nájdete v časti 6.1.

V prípade, že Vami vytvorený zoznam uzlov sa nachádza v adresári s programom, zadajte nasledovný príkaz:

```
$ export MQ_NODES='/home/user/codeploy/zoznam_uzlov'
```

5. Špecifikujte svoj PlanetLab účet pomocou premennej MQ_SLICE nasledovným príkazom: `export MQ_SLICE='cesnet_s2'`. Namiesto sliceu `cesnet_s2` zadajte názov Vášho sliceu.

¹ <http://codeen.cs.princeton.edu/codeploy/>

² <http://codeen.cs.princeton.edu/codeploy/codeploy.tar.gz>

6. Ak sa chystáte prihlasovať k uzlom. Ku ktorým Ste ešte neboli prihlásení, zadajte nasledujúci príkaz:

Príkaz 3.1: Zrušenie overovania kľúču pri prihlasovaní.

```
$ echo "StrictHostKeyChecking no" >> ~/.ssh/config
```

Uvedený príkaz vypne bezpečnostné nastavenie overovania kľúču uzlu pri ssh prihlasovaní. Ak Vám uvedený spôsob nevyhovuje môžete potvrdzovať prihlásenie k uzlom manuálne. Kľúče uzlov PlanetLabu sa občas menia a preto odporúčam zoznam známych uzlov „~/.ssh/known_hosts“ pred každým prihlasovaním zmazať aby Vám prípadná nezhoda kľúčov nebránila v prihlásení.

7. Pre automatické nastavenie premenných programu pre všetky budúce prihlásenia je potrebné pridať do súboru „~/.bashrc“ priradenie cesty k adresáru s programom do premennej PATH spolu s príkazmi z predchádzajúcich bodov. Adresárovú cestu k programu upravte podľa jeho umiestnenia vo Vašom systéme. Uvedené premenné „MQ_NODES“ a „MQ_SLICE“ zadajte len v prípade, že budete dlhodobo pracovať s tým istým sliceom a súborom so zoznamom uzlov.

Príkaz 3.2: Nastavenie premenných prostredia pre program CoDeploy.

```
PATH=$PATH:/home/user/codeploy
export PATH
export MQ_NODES='/home/user/codeploy/zoznam_uzlov'
export MQ_SLICE='cesnet_s2'
```

8. Pomocou príkazu `ssh-add /home/user/.ssh/id_rsa` zadajte ssh prihlasovaciemu agentovi cestu k privátnemu kľúču. Uvedený príkaz upravte podľa umiestnenia a názvu súboru s Vaším kľúčom. Po potvrdení zadajte passphrase, ktorý Ste použili pri vytvorení kľúčového páru. Postup uvedený v tomto bode je dôležitý pretože program CoDeploy predpokladá správne priradenie užívateľovej identity a prístup k uzlom bez zadávania passphrase.

Nastavenia voliteľných premenných prostredia:

- | | |
|------------|--|
| MQ_TIMEOUT | – počet sekúnd k ukončeniu nedokončených požiadaviek, default 30 s. |
| MQ_DELAY | – počet sekúnd medzi spustením dvoch ssh príkazov, default 0 s. |
| MQ_ORDER | – ak je nastavená na hodnotu 1 výstupné výpisy programu sú zoradené podľa poradia, najskôr sada výpisov ssh prihlasovania, potom výsledky vykonania samotného príkazu. |
| MQ_MAXLOAD | – špecifikuje maximálny počet otvorených spojení. |

3.2.4 Inštalácia a nastavenie web serveru

Program CoDeploy vyžaduje na upload súborov ich uloženie na miesto prístupné z internetu. Na splnenie tejto požiadavky postačuje napr. web server Apache. Ďalšie potrebné funkcie vyplývajú z nárokov Vami vytvoreného projektu. Pre potreby praktickej časti uvedenej na konci tejto kapitoly postačuje podpora skriptovacieho jazyka PHP.

Inštalačný balík web serveru Apache2 je dostupný cez repozitáre Ubuntu. Pre jeho inštaláciu zadajte do terminálu príkaz `sudo apt-get install apache2` (použ. v. 2.2.16-1ubuntu3.1). Podporu PHP a odpovedajúci modul pre Apache je možné doplniť inštaláciou odpovedajúcich balíčkov pomocou príkazu `sudo apt-get install php5 libapache2-mod-php5` (použ. v. 5.3.3-1ubuntu9.5).

Po inštalácii PHP5 je možné skripty napísané v tomto jazyku spúšťať cez prehliadač. Základné nastavenie modulov web serveru Apache by malo spúšťanie PHP skriptov podporovať. Skontrolujte či sa nasledujúce súbory modulu PHP nachádzajú v zložke povolených modulov programu Apache:

```
/etc/apache2/mods-enabled/php5.conf  
/etc/apache2/mods-enabled/php5.load
```

V prípade, že uvedené súbory nenájdete použite príkaz `sudo a2enmod php5` na povolenie Apache modulu pre podporu PHP5. Pre správnu funkčnosť povoleného modulu je potrebné web server reštartovať príkazom `sudo /etc/init.d/apache2 restart`.

Pre overenie spúšťania PHP5 skriptov vytvorte súbor s názvom „phpinfo.php“ s týmto obsahom

```
<?php  
    phpinfo();  
?>
```

a uložte ho do koreňového adresáru web serveru Apache, defaultne „/var/www“. Skript musí byť spustiteľný. Pri zadaní odkazu „http://hostname/phpinfo.php“ do prehliadača sa zobrazia hodnoty rôznych konfiguračných parametrov PHP5. Výraz „hostname“ v uvedenom odkaze nahraďte ip adresou alebo názvom Vami vytvoreného web serveru.

3.2.5 Použitie programu CoDeploy

Použitie programu je jednoduché. Stačí zadať tri argumenty spustenia potrebné pre proces prenosu. Jedná sa v podstate o názov lokálneho adresáru, URL zodpovedajúci tomuto adresáru a názov vzdialeného adresáru.

Za predpokladu, že použijete defaultný lokálny adresár „/var/www“ dostupný cez web server pomocou URL „http://hostname/“ a jeho obsah chcete skopírovať na Váš PlanetLab slice do adresára „test“ zadajte nasledujúci príkaz.

Príkaz 3.3: Syntax príkazu programu CoDeploy.

```
$ ./codeploy /var/www http://hostname/ test
```

Aby sme mohli pracovať v adresári „/var/www“, ktorý bol vytvorený užívateľom root, je potrebné prevziať jeho vlastníctvo príkazom `sudo chown -R username /var/www/`. Parameter „-R“ mení rekurzívne aj vlastníka podadresárov. Výraz „username“ nahraďte názvom Vášho užívateľského účtu.

Ak chcete optimalizovať rýchlosť prenosu pri prvom uploade, k príkazu spustenia programu pridajte parameter „-a“. Tým dôjde k pridaniu adresáru do tar balíčku a k jeho následnej kompresii. V prípade ďalších prenosov slúžiacich len na aktualizáciu niektorých zmenených súborov použite prenos bez zadania parametru „-a“.

Pozn.: môžete si všimnúť, že CoDeploy pri prenose vytvorí dočasné súbory. Dôvodom je, že adresár web serveru obsahuje spustiteľné súbory. CoDeploy vytvorí ich dočasnú nespustiteľnú kópiu. Zamedzí sa tým spusteniu týchto súborov namiesto ich skopírovania. Podobne program spracuje aj súbory s nedávnym časom modifikácie a nahradí ho časom starším. Takéto súbory by mohli byť pre distribučnú sieť nespracovateľné.

3.2.6 Použitie nástrojov Multiquery a Multicopy

Napr., pre vzdialené vytvorenie adresáru a spustenie programu na pozadí použijete nástroj multiquery:

Príkaz 3.4: Syntax príkazu programu Multiquery.

```
$ ./multiquery 'mkdir results; directory/program &'
```

Uzatvorenie príkazu do úvodzoviek znamená jeho vykonanie na vzdialenom uzle a znak „;“ slúži na oddelenie dvoch nezávislých príkazov.

Nástroj multicopy môžete použiť na skopírovanie súborov na vzdialený uzol alebo naopak. Tento nástroj používa parameter „@“, ktorý je v prípade, že je uvedený samostatne nahradený názvom vzdialeného uzlu. Ak je uvedený spolu s dvojbodkou tj. „@:“ je nahradený názvom sliceu a uzlu. Príkaz `./multicopy filename @:directory` slúži na skopírovanie súboru do adresáru na všetky vzdialené uzly.

Na stiahnutie súborov z adresáru na vzdialených uzloch a ich skopírovanie do vytvoreného lokálneho adresáru a podadresárov s názvami jednotlivých uzlov použijete nasledujúce príkazy:

Príkaz 3.5: Syntax využívaná programom Multicopy.

```
$ mkdir newdir
$ ./multicopy newdir newdir/@
    Chybu po vykonaní príkazu môžete ignorovať.
$ ./multicopy '@:directory/*' newdir/@
```

Prvý multicopy príkaz slúži na vytvorenie adresárov s názvami jednotlivých vzdialených uzlov vo Vami vytvorenom adresári „newdir“. Druhý príkaz skopíruje súbory z adresáru „directory“ na vzdialených uzloch do odpovedajúcich lokálnych adresárov.

Obidva uvedené nástroje pracujú s rovnakým zoznamom uzlov aký je priradený programu CoDeploy

3.3 Príklad využitia programu CoDeploy

V tejto časti uvediem praktické využitie programu CoDeploy na upload, spustenie a zastavenie nahraného python skriptu. Skript má po spustení za úlohu zistiť a späťne zaslať informácie o názve a polohe organizácie, pod ktorú uzol spadá. Tieto informácie sú následne pomocou php skriptov uložené do textového súboru a prekonvertované do kml súboru pre zobrazenie v aplikácii Google Earth. Pomocou uvedeného postupu je teda možné zistiť polohu všetkých organizácií, ktoré v rámci PlanetLabu poskytujú svoje uzly pre ostatných užívateľov.

Základný návod v úplnom znení je uvedený v literatúre [2], kapitola 5. Nasledujúce časti obsahujú praktické vypracovanie a výstup tejto práce.

3.3.1 Požiadavky

1. Python vo verzií 2.4 alebo vyššej.
2. Adresár s prístupovými právami na zápis a s možnosťou prístupu cez web server.
3. Web server s podporou PHP (≥ 4.1).
4. Služba Google Earth.

3.3.2 Zoznam serverov na upload

Na získanie zoznamu severov pre upload python skriptu som použil aplikáciu CoMon. V tomto prípade som potreboval vytvoriť zoznam obsahujúci jeden aktívny uzol z každej organizácie. Použil som teda tieto vzorové premenné uvedené na *stránke*¹ aplikácie:

<code>format=nameonly</code>	– zobrazí iba názvy uzlov
<code>select='resptime > 0'</code>	– vyberie uzly, ktoré reagujú na požiadavky aplikácie
<code>persite=1</code>	– vyberie jeden uzol z každej organizácie

¹ <http://comon.cs.princeton.edu>

Následne som uvedené premenné pridal do cgi skriptu aplikácie CoMon čím som získal nasledujúci hypertextový odkaz.

```
http://comon.cs.princeton.edu/status/tabulator.cgi?table=table_nodeviewsh  
ort&format=nameonly&persite=1&select='resptime > 0'
```

Zadaním uvedeného odkazu do prehliadača som získal požadovaný zoznam a jeho obsah som skopíroval do textového súboru „nodes“ vytvoreného v adresári programu CoDeploy.

Uzly uvedené v zozname som následne pridal do svojho sliceu pomocou priloženého python skriptu (*Skript 8.1*), ktorý slúži na prístup k programovému rozhraniu PlanetLabu. V prípade jeho úspešného vykonania v terminály posledný príkaz vráti hodnotu „1“. Po zadaní príkazov je potrebné počkať aj niekoľko hodín kým sa vytvoria virtuálne servery na všetkých pridaných uzloch.

3.3.3 Príprava dát na web servery

Pre potreby praktickej ukážky využitia programu CoDeploy som využil nezabezpečený web server Apache so základnými nastaveniami. Treba si ale uvedomiť, že nezabezpečený web server nie je vhodné používať na prenos osobných dát resp. na iné ako testovacie účely. Preto ak poznáte bezpečnostné nastavenia Vášho web serveru dostatočne na to aby ste nakonfigurovali zabezpečený prístup urobte tak. Inštalácia a potrebné nastavenia web serveru boli uvedené v kapitole 3.2.4.

Na získanie dát o polohe organizácií je potrebné pripraviť tieto súbory. Jedná sa o dva formátovacie php skripty „phonelog.php“ (*Skript 8.2*) a „phonehome.php“ (*Skript 8.3*). Ďalším je python skript na získanie potrebných informácií o názve a polohe organizácie „phonehome.py“ (*Skript 8.4*). Obsah premennej „phonehome_url“ v python skripte je potrebné nahradiť odkazom na súbor „phonehome.php“ uloženom na web servery. Posledným súborom je prázdny textový súbor „phonelog.txt“, do ktorého sú údaje zapisované. Pre uvedené súbory je potrebné správne nastaviť prístupové práva. Tri potrebné skripty musia byť spustiteľné a k súboru „phonelog.txt“ musí byť povolený prístup zápisu zasielaných informácií z internetu. Nastavenie práv som vykonal nasledovnými príkazmi:

```
$ chmod +w phonelog.txt  
$ chmod +x phonelog.php phonehome.php phonehome.py
```

3.3.4 Upload, spustenie a zastavenie skriptu

Na servery, ktoré som pridal do svojho sliceu je potrebné uploadovať skript „phonehome.py“. Umiestnil som ho teda do adresáru „/var/www“ a na jeho upload som použil nasledujúci príkaz spustený v adresári programu CoDeploy:

```
$ ./codeploy /var/www http://147.229.213.xxx/ hello
```

Pozn.: posledný argument je názov adresáru na vzdialenom servery, do ktorého sa dáta skopírujú.

Následne som do adresáru web serveru umiestnil ostatné potrebné súbory. Pre spustenie skriptu som použil nástroj Multiquery, ktorým som na jednotlivých uzloch zaregistroval službu cronjob. Táto služba následne spúšťala uploadovaný skript „phonehome.py“ na jednotlivých uzloch každé dve minúty.

```
$ ./multiquery 'echo "*/2 * * * * python hello/phonehome.py" | crontab -;  
sudo /sbin/service crond start'
```

Po ukončení činnosti programu som počkal pár minút aby boli odpovede od serverov kompletné a deamona služby cron som zastavil.

```
$ ./multiquery 'sudo /sbin/service crond stop'
```

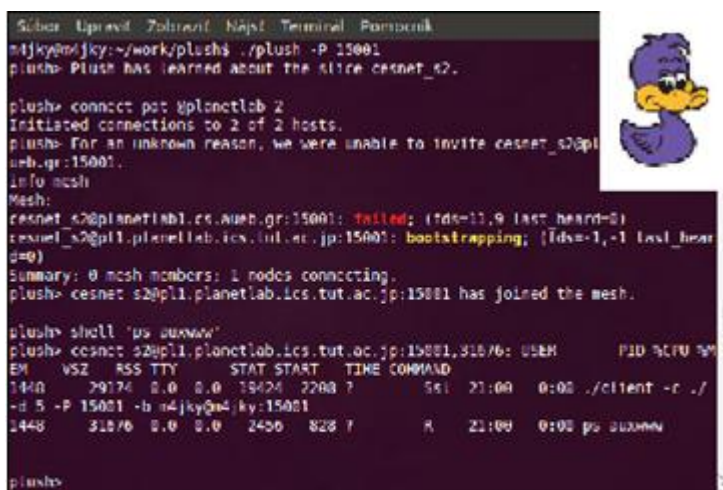
Pozn.: nenechávajte službu spustenú v čase, keď ju nevyužívate, z dôvodu veľkej záťaže pre web server.

3.3.5 Zobrazenie získaných dát

Získané dáta uložené v súbore „phonelog.txt“ je ešte potrebné prekonvertovať. Na konverziu slúži pripravený php skript „phonelog.php“. Stačí ho len spustiť napr. zadáním loopback adresy a cesty k danému skriptu do prehliadača. Zobrazí sa okno ohľadom miesta uloženia už prekonvertovaných dát tj. súboru „hello_world.kml“. Tento súbor je následne možné otvoriť v programe Google Earth, prípadne internetovej službe Google maps.

Aby bolo možné program Google Earth spustiť je potrebné nainštalovať balíček so štandardným systémom pre podporu aplikácií tretích strán `lsb-core` (použ. v. 4.0-0ubuntu8). Priložený výstupný obrázok (Obr. 8.1) som získal pomocou služby Google maps. Dôvodom bola možnosť plochého zobrazenia mapy sveta, ktorú program Google Earth nepodporuje. Súbor kml je možné do služby Google maps načítať ako hypertextový odkaz na súbor „hello_world.kml“ zadaný do poľa pre vyhľadanie miesta. Uložil som ho teda na vytvorený web server a do vyhľadávacieho poľa Google maps som zadal odkaz „http://147.229.213.xxx/hello_world.kml“.

3.4 Plush



Obr. 3.2: Terminálové prostredie Plushu.

Aplikácií na podobnom princípe resp. s rovnakými funkciami je v sieti PlanetLab niekoľko. Hlavným dôvodom výberu nástroja Plush bola jeho grafická nadstavba Nebula, ktorá poskytuje možnosť vývoja vlastnej aplikácie pomocou definovaných objektov. Tento spôsob je vhodný pre nových užívateľov siete pretože nahradzuje prácu v príkazovom riadku. Užívateľ sa môže zamerať na vývoj samotnej aplikácie a o ostatné kroky sa postará program sám.

3.4.1 Popis programu

Hlavným komponentom aplikácie je kontrolér, typicky spustený na užívateľskom počítači a jednoduchý klientský proces, ktorý beží na všetkých uzloch host'ujúcich aplikáciu Plush. Platforma spravovaných klientov nie je limitovaná len na jednu. Aplikačný kontrolér programu Plush môže naraz pracovať so všetkými podporovanými typmi platforiem ako je napr. PlanetLabu, Grid a akékoľvek uzly užívateľovej organizácie.

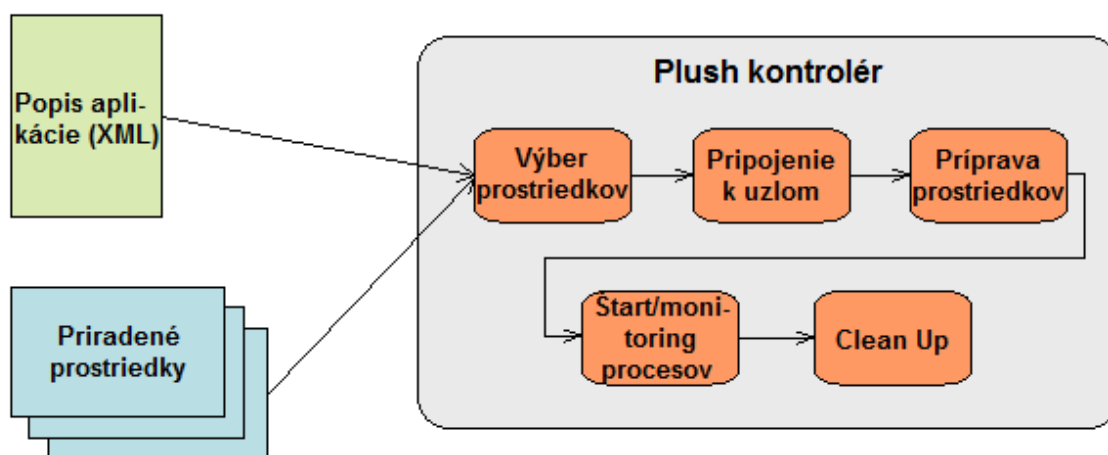
Klient spustený na každom vzdialenom uzle vykonáva príkazy podľa požiadaviek lokálneho kontroléru programu. Napr., kontrolér môže požiadať vzdialených klientov o inštaláciu RPM balíčkov alebo o spustenie a monitoring procesu a po vykonaní požadovaných úloh kontrolér spätne o tejto skutočnosti informovať.

Plush obsahuje samozavádzací mechanizmus, ktorý v prípade potreby umožní inštaláciu klienta na vzdialený uzol. Skript tohto mechanizmu stiahne potrebné aplikačné súbory priamo zo zdroja alebo poskytne zdrojový kód sám a následne klienta na vzdialený uzol nainštaluje. Preposlanie zdrojového kódu programu je v sieti PlanetLab užitočné. Umožňuje vyhnúť sa inštalácií potrebných knižníc a zaťaženiu uzlov pri kompilácii klienta a prejsť priamo k inštalácii staticky priradených klientských aplikácií.

Cieľom nástroja Plush je poskytnúť rozšíriteľný model správy pre veľké distribučné siete. Užívateľ charakterizuje a poskytne spracovanie projektu vo formáte XML. Na základe tohto dokumentu Plush pre užívateľa lokalizuje, kontaktuje a pripraví potrebné prostriedky. Následne spustí danú aplikáciu, monitoruje prípadné chyby pri spustení, spravuje údržbu po ukončení procesu a vykonáva ďalšie súvisiace funkcie.

3.4.2 Plush kontrolér

V prípade typického spúšťania kontrolér využíva xml popis aplikácie a zoznam dostupných prostriedkov (napr. názvy sliceov PlanetLabu a ostatných počítačov dostupných cez SSH). Na základe požiadaviek aplikácie vyhľadá a priradí potrebné prostriedky. Následne sa kontrolér na vybrané uzly pripojí, nainštaluje sadu užívateľsky definovaných softvérových balíčkov a skopíruje súbory projektu na tieto uzly. Po dokončenej príprave Plush nakonfiguruje procesy, ktoré budú na uzloch spustené. Všetky uzly sa synchronizujú pomocou blokovacieho synchronizačného bodu. Po dokončení synchronizácie sú procesy spustené. Ak je proces ukončený alebo dôjde k inej užívateľom definovanej akcii označujúcej ukončenie experimentu, sú na vzdialených uzloch spustené tzv. čistiace úkony. Uvedené kroky sú zobrazené na nasledujúcom obrázku.



Obr. 3.3: Popis funkcie Plush kontroléru.

3.4.3 Požadované knižnice

Ďalej uvedený postup bol vypracovaný pre Ubuntu 10.10 x64. V prípade, že využivate inú distribúciu použite ako oporu pôvodný *návod*¹ uverejnený na web stránke Plushu. Neinštalujte program do virtuálneho stroja ak sa chystáte využívať GUI Nebula. Grafické rozhranie vyžaduje správnu funkčnosť grafického prostredia opengl, ktoré v aktuálnej verzii VirtualBoxu (4.0.6) nefunguje v prípade Nebuly korektne.

Pred samotnou inštaláciou potrebných knižníc si nainštalujte základný balíček vývojových nástrojov.

¹ <http://plush.cs.williams.edu/users.html>

Príkaz 3.6: Príkaz na inštaláciu knižnice.

```
$ sudo apt-get install build-essential (použ. v. 11.5)
```

Plush vyžaduje tieto knižnice (názvy knižníc určených na inštaláciu sú zvýraznené fontom Courier): balíčky použitých verzií knižníc sú priložené na médiu.

- **libreadline:** knižnica podporujúca konzistentnosť užívateľského rozhrania medzi samostatnými programami využívajúcimi rozhranie príkazového riadku. Taktiež poskytuje rozhranie na vyvolanie histórie vstupných užívateľských príkazov. Knižnica `libreadline6` je defaultne nainštalovaná ale je potrebné nainštalovať jej vývojový balíček `libreadline6-dev` (použ. v. 6.1-3) spolu, s ktorým sa nainštaluje aj knižnica `libncurses5-dev`. Základný balík `libncurses5` (použ. v. 5.7+20100626-0ubuntu1) obsahujúci zdieľané knižnice potrebné na prevádzku programov skompilovaných s „ncurses“ je defaultne nainštalovaný.
- **xmlrpc-c:** XML-RPC protokol je rýchla a jednoduchá cesta ako vykonávať volania procedúr cez internet. Pomocou neho je volanie procedúry prekonvertované do XML dokumentu, odoslané na vzdialený server použitím HTTP a následná odpoveď je taktiež prevedená do formátu XML. Knižnica poskytuje štandardnú podporu protokolu XML-RPC pre jazyky C a C++, obmedzené prostriedky pre využitie protokolu v kombinácii s HTTPS a variáciu protokolu využívajúcu streamovanie paketov.

Pred jej inštaláciou je potrebné nainštalovať nasledovné knižnice: `libcurl4-openssl-dev` (použ. v. 7.21.0-1ubuntu1), `autoconf` (použ. v. 2.67-2ubuntu1), `texinfo` (použ. v. 4.13a.dfsg.1-5ubuntu1). Samotná knižnica `xmlrpc-c` (použ. v. stable, revision 2132) je dostupná cez subversion archív. Na prístup k tomuto archívu je potrebné nainštalovať knižnicu klienta `subversion`. Jedná sa o náhradu CVS, systému na sledovanie zmien v kóde programu. Nasledujúce príkazy zadajte v adresári, do ktorého chcete knižnicu stiahnuť:

Príkaz 3.7: Download a inštalácia knižnice xmlrpc.

```
$ REPOS=http://xmlrpc-c.svn.sourceforge.net/svnroot/xmlrpc-c/stable
$ svn checkout $REPOS xmlrpc-c
$ cd xmlrpc-c
$ ./configure
$ make
$ sudo make install
```

Pozn.: ak chcete stiahnuť najnovšiu verziu nahraďte v http odkaze výraz „stable“ za výraz „advanced“. V tomto návode bola použitá verzia stable.

- **libcurl:** knižnica určená na multiprotokolový prenos súborov. Knižnica `libcurl3` (použ. v. 7.21.0-1ubuntu1) je defaultne nainštalovaná.
- **libxml:** poskytuje rozsiahle programové rozhranie na vytvorenie a spracovanie XML dátových súborov. Knižnica `libxml2` (použ. v. 2.7.7.dfsg-4ubuntu0.1) je defaultne nainštalovaná.
- **zlib:** knižnica obsahujúca kompresnú metódu pre program `gzip` a `pkzip`. Knižnica `zlib1g` (použ. v. 1:1.2.3.4.dfsg-3ubuntu1) je defaultne nainštalovaná.
- **libc:** balíček obsahujúci štandardnú knižnicu jazyka C, štandardnú matematickú knižnicu a mnoho ďalších. `libc6` (použ. v. 2.12.1-0ubuntu10.2) je defaultne nainštalovaný. V balíčkuje zahrnutá aj podpora dynamického linkovania a programové rozhranie „`pthread`“ umožňujúce aplikáciám vytvárať a spravovať vlastné vlákna.
- **libssl:** defaultne nainštalovaná zdieľaná knižnica `libssl` (použ. v. 0.9.8o-1ubuntu4.4) potrebná napr. pre programy `openssh`, `apache-ssl` a `telnet-ssl`. Súčasťou je aj potrebná knižnica `libcrypto`.
- **libidn:** defaultne nainštalovaná knižnica `libidn11` (použ. v. 1.18-1) slúžiaca na kódovanie a dekódovanie medzinárodných doménových názvov.
- **libares:** knižnica pre asynchrónne vyhľadávanie DNS. Inštalácia knižnice `libares0` (použ. v. 1.1.1-5.1build1) je nepovinná ale odporúčaná pre zlepšenie výkonu v sieti PlanetLab.
- **lex, yacc:** na generovanie slovníkovej analýzy textu posluží nástroj `flex` (použ. v. 2.5.35-9.1) a ako syntaktický generátor kompatibilný s jazykom `yacc` program `bison` (použ. v. 1:2.4.1.dfsg-3).

Plush využíva niekoľko pomocných skriptov napísaných v jazyku Perl. Ak chcete aby spolupracoval s centrálnou databázou PlanetLabu je potrebné správne nainštalovať nástroje `Frontier::Client` (použ. v. `Frontier-RPC-0.07b4p1`) a `Crypt::SSLeay` (použ. v. `Crypt-SSLeay-0.58_01`). Ich inštalácií sa venuje časť 3.4.5.

3.4.4 Inštalácia a kompilácia

V prípade, že používate distribúciu Ubuntu 10.10 x64 alebo jej 32-bitovú verziu môžete využiť staticky skompilované zdrojové súbory programu. Tar balíček pre 64-bitový systém

stiahnete tu^1 a pre 32-bitový tu^2 . V tejto práci bola overená funkčnosť obidvoch aktuálne dostupných verzií. Pokiaľ Vám 64-bitová verzia korektne nefunguje môžete v 64-bitovom systéme vyskúšať aj 32-bitovú variantu. V prípade, že dôjde k zmene zdrojových súborov programu a skompilované balíčky už budú nefunkčné môžete skúsiť aktualizovať pomocné skripty programu Plush, nachádzajúce sa v adresári „plush/helper-scripts“, súbormi dostupnými na servery *SourceForge*³. Ak uvedená metóda nepomôže budete musieť program samostatne skompilovať.

V prípade, že využívate inú distribúciu, v ktorej nefunguje korektne ani jeden z uvedených skompilovaných balíčkov, musíte si stiahnuť zdrojové súbory programu a kompiláciu vykonať samostatne. Pred samotnou kompiláciou je potrebné nainštalovať ďalšie knižnice. **Ich názvy budú v nasledujúcom texte opäť znázornené fontom Courier a použité verzie nájdete na priloženom médiu.** Názvy nasledujúcich knižníc ako aj knižníc potrebných pre samotnú aplikáciu Plush sa môžu v inej distribúcii líšiť a pre správnu kompiláciu môže byť nevyhnutné ručne upraviť súbor „Makefile.in“.

Zoznam potrebných knižníc:

1. `libxml2-dev` (použ. v. 2.7.7.dfsg-4ubuntu0.1): vývojové súbory XML.
2. `libboost-all-dev` (použ. v. 1.42.0.1ubuntu1): vývojové prostredie Boost.
3. `cvs`: nástroj na aktualizáciu softvéru.
4. `xutils-dev` (použ. v. 1:7.5+4): balíček obsahujúci mimo iného potrebný vývojový nástroj `makedepend` slúžiaci na vytvorenie závislostí pre súbory `makefile`.
5. `libunistring0`, `libunistring-dev` (použ. v. 0.9.3-1): knižnica unicode reťazcov pre jazyk C a jej vývojové súbory.

Plush je dostupný pomocou nástroja `cvs` na servery *SourceForge*. Nasledujúce príkazy zadajte do terminálu. Ak potrebujete program skompilovať dynamicky odstráňte obidva parametre „`-with-static`“.

Príkaz 3.8: Download a kompilácia programu Plush.

```
$ cvs -d:pserver:anonymous@plush.cvs.sourceforge.net:/cvsroot/plush login
Heslo nevyplňujte a potvrdzte klávesov ENTER.
$ cvs -z3 -d:pserver:anonymous@plush.cvs.sourceforge.net:/cvsroot/plush
checkout plush
$ cd plush
$ autoconf
```

¹ <http://plush.cs.williams.edu/plush-binaries/plush-bin64-ubuntu10.tar.gz>

² <http://plush.cs.williams.edu/plush-binaries/plush-bin.tar.gz>

³ <http://plush.cvs.sourceforge.net/viewvc/plush/plush/helper-scripts/?sortdir=down>

```
$ ./configure --with-static-client --with-static-plush
$ make dep
$ make
```

Po úspešnej kompilácii by mali byť v adresári programu vytvorené binárne súbory „plush“ a „client“.

3.4.5 Nastavenie užívateľského prostredia

Program Plush vyžaduje prihlasovanie bez overovania pravosti host'ovského kľúča. Aby nebolo v priebehu prihlasovania požadované potvrdenie je potreba zadať nasledujúci príkaz:

```
$ echo "StrictHostKeyChecking no" >> ~/.ssh/config
```

Uvedený príkaz je dôležitý aj pre proces porovnávania kľúča uzlu s kľúčom uloženým v súbore „~/.ssh/known_hosts“. Kľúče uzlov sa občas menia čo v prípade ich nezhody predstavuje ďalšiu prekážku pri prihlasovaní. Aby Ste pri prihlasovaní k viacerým uzlom nemuseli pri každom SSH prihlasovaní zadávať passphrase priradiť cestu k privátnemu kľúčovi prihlasovaciemu agentovi pomocou príkazu `ssh-add /home/user/.ssh/id_rsa`, pričom uvedenú cestu upravte podľa umiestnenia Vášho kľúča. Po potvrdení zadajte passphrase, ktorý bol použitý na jeho vytvorenie.

Plush obsahuje tzv. adresár uzlov. Je to komponent, ktorý zahŕňa všetky známe zdroje. Každý slice a uzol PlanetLabu ako aj ostatné uzly, ku ktorým máte prístup a chcete aby ich Plush využíval, musia byť do tohto adresáru pridané. Samozrejme ku všetkým pridaným uzlom musíte mať možnosť pristupovať cez SSH pomocou kľúčového páru alebo inej neinteraktívnej metódy prihlasovania. V tomto návode bol na tieto účely využitý ssh-agent.

Plush môže byť tiež využitý na prácu s databázou PlanetLabu ako napr. automatické obnovenie sliceu, pridanie a odobranie uzlov sliceu a na overenie jeho stavu. Prihlásenie k databázovému serveru vyžaduje prihlasovacie meno a heslo k účtu vytvorenému na stránke PlanetLabu. Pre nastavenie prihlasovacích údajov ako aj ďalších nastavení programu Plush zadajte v adresári programu nasledujúci príkaz:

Príkaz 3.9: Spustenie konfiguračného skriptu programu Plush.

```
$ perl helper-scripts/setup.pl
```

Program nastavte nasledovne: popis jednotlivých položiek a hlásenia konfiguračného skriptu boli vynechané.

Príkaz 3.10: Nastavenie programu Plush.

```
Please select from the following options:
 1. Configure known hosts
 2. Configure Plush preferences
 3. Quit
> 1
```

```

Do you have a PlanetLab account (Y/N) [N]? Y
Please enter your planet-lab.org login: login@stud.feec.vutbr.cz
    Zadaťte Váš login.
Please enter your planet-lab.org password: ****
Please confirm your planet-lab.org password: ****
Please enter your PlanetLab slice: cesnet_s2
    Zadaťte názov Vášho sliceu.
Please choose a port for this slice (use a different port for each
slice): 15001
    Zadaťte iný port pre každý slice.
Please enter your PlanetLab slice:
    Zadaťte názov nasledujúceho sliceu, ktorý chcete používať. Ak Ste
    zadali všetky potvrdte prázdne.
Enter a hostname of the form host:port (empty to finish):
    Ak chcete pridať ďalšie uzly zadaťte ich názvy vo formáte host:port,
    ak nie potvrdte prázdne.
Done...
> 2
Enter your preferred ClientPath (i.e., /home/ucsd_plush/): ./
Enter your preferred ClientPrefix (i.e., ./): ./
Do you plan to use PlanetLab? (Y/N) [Y]: Y
Done...
> 3

```

Varovanie: zadané prihlasovacie heslo je uložené v skrytom súbore „passwd“ v nešifrovanej forme v adresári programu Plush.

Na pripojenie k databáze sú potrebné perl moduly Frontier::Client a Crypt::SSLeay. Nasledujúcim príkazom spustíte v adresári programu Plush testovací PlanetLab skript, pričom výraz „login@stud.feec.vutbr.cz“ nahraďte Vašou PlanetLab email adresou resp. loginom.

Príkaz 3.11: Spustenie testovacieho skriptu programu Plush.

```
$ helper-scripts/handle-plcdb.pl login@stud.feec.vutbr.cz x x listall
```

Ak uvedený príkaz vypíše hlásenie o chýbajúcich moduloch využite na ich inštaláciu nástroj cpan a zadaťte nasledujúce príkazy:

Príkaz 3.12: Inštalácia modulov Frontier::Client a Crypt::SSLeay nástrojom cpan.

```

$ sudo cpan
Would you like me to configure as much as possible automatically? [yes]
    Potvrdte ak chcete aby sa nástroj nakonfiguroval automaticky.
$ install Frontier::Client
...
$ install Crypt::SSLeay
...
Which SSL install path do you want to use? [/usr]
    Potvrdte alebo zadaťte inú cestu.
Do you want to run the live tests (y/N)? [N]

```

Potvrďte a po ukončení cpan pre overenie opäť použite testovací
Skript.
\$ quit

V prípade správnej inštalácie sa po spustení testovacieho skriptu zobrazí výpis uzlov Vášho sliceu.

3.4.6 Spustenie

Pre spustenie programu Plush zadajte „./plush“ v adresári programu. Základné rozhranie programu sa snaží napodobňovať samostatného príkazového interpreta. Väčšina príkazov je ale asynchrónna takže rozlíšenie formátovania popredia/pozadia nie je na úrovni klasického shellu. Pre overenie funkčnosti programu zadajte nasledujúce príkazy:

Príkaz 3.13: Overenie funkčnosti programu Plush.

```
$ ./plush -P 15000
    Spustí Plush na porte 15000.
    Počkajte kým Plush vypíše správu s názvami Vašich sliceov.
plush> connect pat @planetlab 10
    Pripojí sa k desiatim náhodným uzlom, ktorých názov obsahuje výraz
    "planetlab".
    Počkajte kým uvidíte správu o pripravení uzlu resp. jeho pripojenie
    do „mesh“.
plush> info mesh
    Vypíše aktuálny stav pripojenia jednotlivých uzlov.
plush> shell "ps aux"
    Spustí „ps aux“ na každom pripojenom uzle a výsledok vypíše na
    obrazovku.
plush> disconnect
    Odpojí uzly zo zoznamu pripravených prostriedkov a tým zruší ich
    vzdialených klientov
plush> quit
```

3.4.7 Využitie

V adresári „tests“ sa nachádza niekoľko popisov aplikácií v jazyku XML, ktoré môžu byť použité na experimentovanie s rôznymi vlastnosťami programu. Spusteniu vlastného experimentu sa venovať nebudem z dôvodu dostupnosti grafického rozhrania Nebula. Popisu tohto rozhrania sa venuje časť 4.

V tejto časti uvediem aspoň sadu príkazov pre program Plush s vysvetlivkami. V prípade záujmu o využitie programu pre prácu na vlastnom experimente využite *stránku*¹ s rôznymi vzorovými aplikáciami.

Zoznam príkazov:

- `plush> load path:` načíta súbor s projektom.
- `plush> connect hostname:` spustí Plush klienta na vzdialenom uzle.
- `plush> connect slice názov_sliceu:` pripojiť ku všetkým uzlom sliceu.
- `plush> disconnect:` ukončí všetky otvorené spojenia.
- `plush> disconnect hostname:` odpojí sa od uzlu.
- `plush> info control:` vypíše informácie o stave Plush kontroléru.
- `plush> info nodes:` vypíše súhrnné informácie o všetkých známych uzloch.
- `plush> info node hostname:` vypíše informácie o uzle.
- `plush> info slices:` vypíše súhrnné informácie o všetkých známych sliceoch.
- `plush> info slice názov_sliceu:` vypíše informácie o slice.
- `plush> info mesh:` vypíše zoznam priradených uzlov.
- `plush> info rcp:` vypíše súhrnné informácie o čakajúcich prenosoch sôborov.
- `plush> info rcp #:` vypíše informáciu o špecifikovanom prenose.
- `plush> info install:` vypíše súhrnné informácie o čakajúcich inštaláciách.
- `plush> info install #:` vypíše informácie o špecifikovanej inštalácii.
- `plush> run:` spustí prvý experiment aktívneho projektu.
- `plush> shell "príkaz":` spustí príkaz na pripojených uzloch.
- `plush> sync:` požiada všetky pripojené uzly o aktualizáciu svojho stavu.
- `plush> sync hostname:` požiada uzol o aktualizáciu informácií o jeho stave.
- `plush> slice update názov_sliceu:` stiahne aktuálne informácie o slice z PlanetLab databázy.
- `plush> slice list názov_sliceu:` zobrazí informácie o uvedenom slice.

¹ <http://plush.cs.williams.edu/example.html>

- `plush> slice add názov_sliceu`: priradí všetky uzly PlanetLabu k sliceu.
- `plush> slice add názov_sliceu hostname`: priradí uvedený uzol ku sliceu.
- `plush> slice remove názov_sliceu`: odstráni všetky pridané uzly zo sliceu.
- `plush> slice remove názov_sliceu hostname`: odstráni uzol so sliceu.
- `plush> slice reset názov_sliceu`: odstráni všetky pridané uzly zo sliceu.
- `plush> slice renew názov_sliceu`: obnoví uvedený slice.
- `plush> select experiment`: povie Plushu aby pripravil defaultný experiment a konfiguráciu načítaného projektu.
- `plush> quit`: ukončí Plush.

4 GRAFICKÁ NADSTAVBA NEBULA



Obr. 4.1: Grafické prostredie Nebuly.

Nebula je nadstavbové grafické rozhranie pre program Plush umožňujúce užívateľovi vytvoriť, zobrazíť a spustiť aplikáciu bez nutnosti použitia prostredia príkazového riadku. Spolupracuje s nástrojom Plush pomocou protokolu XML-RPC a obsahuje väčšinu jeho funkcií.

4.1 Inštalácia

Program môžete stiahnuť *tu*¹. Nebula vyžaduje funkčnú inštaláciu nástroja Plush a Javu vo verzií 1.6 alebo novšiu. V distribúcií Ubuntu je Java defaultne nainštalovaná. V prípade, že tomu tak nebude, nainštalujte najnovšiu verziu dostupnú v repozitári alebo ju stiahnite na tejto *stránke*².

Nebula vyžaduje knižnicu JOGL pre podporu opengl v Jave. Táto knižnica obsahuje súbor s príponou „.jar“ a niektoré základné knižnice, ktoré Nebula potrebuje. Aktuálna verzia Nebuly obsahuje JOGL pre Linux/x86 a Windows. Pre distribúciu Ubuntu 10.10 x64 je potrebné stiahnuť *túto*³ verziu. Ak využívate iný operačný systém skontrolujte archív dostupných verzií na tejto *stránke*⁴ či neobsahuje aj tú, ktorú potrebujete.

Po rozbalení balíčka JOGL skopírujte všetky súbory s príponou „.so“ z adresáru „lib“ do adresáru „amd64“, ktorý je potrebné vytvoriť v rozbalenom adresári Nebuly. Ďalej je potrebné nahradiť súbor „jogl.jar“ v adresári „lib“, ktorý sa nachádza v koreňovom adresári Nebuly, rovnomenným súborom z adresáru „lib“ knižnice JOGL. Názov adresáru „amd64“ je potrebné zadať do premennej „ARCH“ v run skripte programu Nebula. Namiesto „ARCH=i386“ zadajte „ARCH=amd64“. V prípade, že sa vo Vašej verzii

¹ <http://plush.cs.williams.edu/nebula/nebula.tar.gz>

² <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

³ http://download.java.net/media/jogl/builds/archive/jsr-231-1.0.0/jogl-1_0_0-linux-amd64.zip

⁴ <http://download.java.net/media/jogl/builds/archive/jsr-231-1.0.0/>

knižnice JOGL nachádzalo viac súborov s príponou „.jar“ uistite sa, že sa cesta k nim nachádza za parametrom „-cp“ príkazu java. Ak potrebujete JOGL nainštalovať na iné miesto stačí správne upraviť cestu k potrebným triedam javy a premennú „LD_LIBRARY_PATH“. Keďže Nebula využíva opengl je potrebné povoliť hardvérovú akceleráciu pre grafickú kartu.

V aktuálnej verzii Nebuly je staršia verzia SSH klienta Mindterm, ktorá v distribúcií Ubuntu nefunguje správne. Preto ak Vám nebude správne fungovať SSH prihlasovanie stiahnite si jeho najnovšiu verziu z *tejto*¹ stránky. Po rozbalení stiahnutého archívu nahraďte súbor pôvodného klienta „mindterm.jar“ v adresári „nebula/lib“ jeho aktuálnou verziou. V prípade, že najnovšia verzia nebude fungovať korektne tak verziu 3.4, ktorej funkčnosť bola overená, nájdete na priloženom médiu.

4.2 Konfigurácia

Nebula podobne ako predchádzajúce programy vyžaduje možnosť prihlásenia na uzol bez zadávania passphrase. Opäť je teda potrebné pred jej spustením zadať cestu k privátnemu kľúču prihlasovaciemu agentovi pomocou príkazu „ssh-add ~/.ssh/id_rsa“ a následne zadať passphrase použitý na jeho vygenerovanie. Cestu k privátnemu kľúču upravte podľa jeho umiestnenia a názvu vo Vašom systéme.

Spustite Nebulu pomocou príkazu „./run“ zadanom v jej koreňovom adresári. Po spustení choďte na lištu a otvorte položku Edit → Preferences. V záložke „Plush“ nastavte v položke „Slice“ názov Vášho sliceu, napr. „cesnet_s2“ a položku „Plush location“ nastavte podľa umiestnenia Plush klienta, napr. „/home/user/plush/.plush“. V zadanej ceste k Plush klientovi sa nemôže nachádzať znak „~“ a na konci adresárovej cesty musí byť uvedený spustiteľný súbor klienta aj so spôsobom jeho spustenia, tzn. „./plush“. V prípade potreby môžete tiež zmeniť pôvodnú hodnotu pre „Plush port“ a „XML-RPC callback server URL“. Pôvodné nastavenia sú ale postačujúce. V nasledujúcej záložke SSH nastavte umiestnenie Vášho privátneho kľúču, napr. „/home/user/.ssh/id_rsa“, v položke „Private key location“, pričom opäť zadajte kompletnú cestu. Do nasledujúceho poľa zadajte passphrase, ktorá bola použitá na zabezpečenie zadaného privátneho kľúču.

¹ <http://tech.cryptzone.com/downloads/mindterm.html>

4.3 Návod na použitie

4.3.1 Monitorovanie uzlov PlanetLabu

1. Spustíte aplikáciu Nebula. Po spustení sa Vám zobrazí základné zobrazenie mapy sveta (Obr. 9.1).
2. Na lište v položke „Plush“ vyberte „Start Plush“.
3. Po spustení nástroja Plush a obdržaní potrebných údajov z centrálnej databázy PlanetLabu sa na mape zobrazí poloha jednotlivých organizácií.
4. Zobrazenie je možné priblížiť kolieskom myši, podržaním LMB a presunutím kurzoru cez požadovanú časť mapy sa zobrazí okno s detailnejším zobrazením zvolenej časti. Posun mapy je možný pridržením RMB a pohybom myši.
5. Na monitorovanie uzlov PlanetLabu Nebula poskytuje grafické zobrazenie stavu, v ktorom je na výber sledovanie nasledujúcich veličín.

%CPU Usage — percentuálne vytťaženie CPU

%Mem Usage — percentuálne využitie pamäte

TX1 — priemerný objem odchádzajúcich dát za poslednú minútu

TX15 — priemerný objem odchádzajúcich dát za posledných 15 minút

RX1 — priemerný objem prichádzajúcich dát za poslednú minútu

RX15 — priemerný objem prichádzajúcich dát za posledných 15 minút

6. Dvojitým kliknutím na zobrazenú polohu organizácie spustíte jej monitorovanie. Zobrazí sa Vám nasledujúce okno (Obr. 9.2). Toto okno zobrazuje stav všetkých uzlov organizácie a graf zvolenej veličiny. Ak chcete monitorovať konkrétny uzol, zobrazte si požadovanú oblasť podľa bodu 4 a dvojklikom vyberte požadovaný uzol.
7. Ak chcete na všetkých uzloch sledovať jeden konkrétny parameter chodíte do menu „Plush“ na lište, zvolíte položku „View Load Metric“ a vyberte požadovaný parameter. Farba zobrazenia polohy jednotlivých organizácií sa začne meniť. Zelená zobrazuje organizácie, ktorých veľkosť sledovaného parametru zodpovedá nízkej hodnote, červená naopak vysokej (Obr. 9.3). Na obrázku môžete vidieť grafické zobrazenie percentuálneho zaťaženia CPU zodpovedajúce hodnotám z okna monitoringu v predošlom bode.

4.3.2 Zobrazenie uzlov sliceu

Uzly PlanetLabu je možné v programe Nebula zobraziť dvoma spôsobmi:

1. Zobrazenie všetkých uzlov (pôvodné nastavenie). Do tohto zobrazenia je možné spätne prepnúť pomocou položky Plush → View All Hosts na lište.

2. Do zobrazenia uzlov patriacich k nastavenému sliceu sa dá prepnúť pomocou položky Plush → View Slice Hosts.

4.3.3 Správa sliceu

1. Spustíte aplikáciu Nebula a nástroj Plush (menu Plush → Start Plush).
2. K sliceu môžete uzly pridať alebo ich z neho odobrať. Voľbu pridania uzlov vyberte pomocou položky Plush → Add Hosts to Slice, pre pridanie nových uzlov musíte byť samozrejme prepnutý do režimu „View All Hosts“. Analogicky môžete uzly zo sliceu odobrať pomocou položky „Remove Hosts from Slice“ v režime zobrazenia uzlov sliceu „View Slice Hosts“. Položky pridania a odstránenia uzlov zostanú po ich voľbe aktivované a preto je ich potreba po použití opätovnou voľbou vypnúť.

Po vykonaní zmien v PlanetLab databáze je potrebné počkať aj niekoľko hodín aby došlo k aplikovaniu zmien aj na zvolených uzloch.

3. Uzly, ktoré chcete pridať vyberiete tak, že pomocou LMB a následným potiahnutím myši vyvoláte detailný záber oblasti organizácie, v ktorej sa uzly nachádzajú.
4. Zobrazí sa okno so zoznamom všetkých uzlov zvolenej organizácie/organizácií (Obr. 9.4).
5. Kliknutím na checkbox zobrazený vedľa názvu uzlu môžete zrušiť priradenie zvoleného uzlu k Vášmu sliceu.
6. Výber potvrdíte stlačením tlačidla OK. Ak nechcete pridať žiadny z vybraných uzlov stlačte tlačidlo Cancel.
7. Ak chcete pridať uzly ďalších organizácií opakujte postup podľa bodov 4-7 až pokiaľ k Vášmu sliceu nepriradíte všetky požadované uzly.
8. Nebula poskytuje aj možnosť predĺženia platnosti sliceu cez položku Plush → Renew Slice.

4.3.4 SSH prihlasovanie k uzlom

1. Spustíte aplikáciu Nebula a nástroj Plush (menu Plush → Start Plush).
2. Prepnete sa do záložky „Host View“, v ktorej je zobrazený zoznam všetkých uzlov vo Vašom slice.
3. Pomocou LMB vyberte uzol, ku ktorému sa chcete prihlásiť. Stlačením RMB vyvoláte menu s možnosťou SSH prihlásenia (Obr. 9.5). Ako vidíte podľa položiek menu, z uzla je možné následne odhlásiť prípadne je ho možné označiť ako nefunkčný, aby Ste vedeli, ku ktorému uzlu sa nemáte aktuálne pripájať.

4. Po zvolení možnosti „SSH to <host>“ dôjde k pridaniu novej záložky s SSH terminálom k zvolenému uzlu.

4.3.5 Vytvorenie a spustenie „HelloWorld“ aplikácie

Cieľom tejto časti je popísať postup vytvorenia a spustenia jednoduchej Plush aplikácie, ktorá vytvorí súbor „HelloWorld.txt“ na dvoch uzloch PlanetLabu.

1. Spustíte aplikáciu Nebula a nástroj Plush (menu Plush → Start Plush).
2. Prepnete sa do záložky „Application View“, ktorá slúži na vytvorenie Plush aplikácie a jej spustenie.
3. Stlačte tlačidlo „Component“ na vytvorenie nového prvku aplikácie. Vytvorený objekt je abstraktným balíkom zdrojov pre Vašu aplikáciu. V tejto časti návodu sa zameriame na vytvorenie objektu, ktorý bude obsahovať dva uzly PlanetLabu.
4. V otvorenom okne „Add Component Block“ vyplňte potrebné položky podľa prílohy (Obr. 9.6) a potvrdte zmeny stlačením OK. Uistite sa, že ste zadali názov sliceu, ktorý chcete použiť.
5. Novo vytvorená položka sa zobrazí na ploche. Kliknite na vytvorený objekt a stlačte tlačidlo „Process“ na pridanie nového procesu.
6. Editačné okno procesu vyvoláte dvojité kliknutím na vytvorený proces. Položky otvoreného okna vyplňte podľa prílohy (Obr. 9.7) a zmeny potvrdte stlačením OK. Vytvorený proces vykoná na priradených uzloch príkaz `touch "HelloWorld.txt"`.
7. V tomto bode by malo byť zobrazenie Vašej Plush aplikácie totožné s priloženým Obr. 9.8.
8. Vytvorenú Plush aplikáciu uložte voľbou „Save As“ v menu „File“ na hornej lište alebo stlačením klávesovej skratky Ctrl + S.
9. Aplikácia je pripravená na spustenie. Môžete ju spustiť kliknutím na tlačidlo „Run Application“.

4.3.6 Používanie SWORD vyhľadávania prostriedkov

Cieľom tejto časti je zoznámenie s integrovaným vyhľadávaním SWORD poskytovaným aplikáciou Plush/Nebula. SWORD je nástroj na vyhľadávanie zdrojov v sieti PlanetLab. Umožňuje vyhľadanie uzlov, ktoré spĺňajú zadané požiadavky. Viac informácií o nástroji

SWORD nájdete *tu*¹. Príklady vyhľadávania pomocou nástroja SWORD nájdete v sekcii *FAQ*².

Uvediem príklad pre upload aplikácie na uzly s nízkou odozvou:

1. Vytvorte nový komponent pomocou návodu v časti 4.3.5. V okne „Add Component Block“ kliknite na tlačidlo „More>>“. Zobrazí sa okno s rozšírenými nastaveniami pre SWORD vyhľadávanie.
2. V menu „Attribute“ vyberte položku „resptime“. Ostatné položky vyplňte podľa prílohy (Obr. 9.9).
3. Predchádzajúce nastavenia nakonfigurujú Plush tak aby vyberal len uzly s maximálnou uprednostňovanou odozvou do 2 sekúnd a absolútnou max. požadovanou odozvou do 3 s.

Bližší popis významu jednotlivých parametrov:

Absolute min – požadovaná spodná hranica žiadanej hodnoty
Absolute max – požadovaná horná hranica žiadanej hodnoty
Ideal min – uprednostňovaná spodná hranica žiadanej hodnoty
Ideal max – uprednostňovaná horná hranica žiadanej hodnoty
Penalty – počet abstraktných penalizačných bodov udelených za odchýlku hodnoty atribútu ak je aktuálna hodnota jednotky v rozmedzí „Absolute min“ a „Ideal min“ alebo „Ideal max“ a „Absolute max“. Detailnejší popis parametrov nájdete *tu*³.

4. Stlačením tlačidla „Add/Update“ priradíte parametre vyhľadávania SWORD k Vašej Plush aplikácii.
5. Potvrďte zmeny tlačidlom OK.

Ďalšími podporovanými parametrami pre SWORD vyhľadávanie zdrojov sú:

oneminload – počet aktívnych procesov za poslednú minútu
fiveminload – počet aktívnych procesov za posledných 5 minút
uptime – doba nepretržitej prevádzky uzlu
cpuspeed – takt procesoru v GHz
busycpu – percentuálne využitie procesorového času
syscpu – percentuálne využitie procesorového času systémom

¹ <http://sword.cs.williams.edu>

² <http://plush.cs.williams.edu/nebula/faq.html#1>

³ <http://sword.cs.williams.edu/querylang.html>

freecpu	– voľný procesorový čas v percentách
memsize	– veľkosť operačnej pamäte v GB
freemem	– voľná operačná pamäť
disksize	– kapacita pevného disku v GB
diskused	– percentuálne využitie disku
txrate	– priemerný objem odoslaných dát v kbps
rxrate	– priemerný objem prijatých dát v kbps

Pre vyhľadanie „dobrých“ zdrojov v sieti PlanetLab je vhodné SWORD premennú „resptime“ použiť spoločne s premennou „oneminload“ a jej parametre v okne SWORD vyhľadávania nastaviť na hodnoty „0, 0, 0, 5, 1“, v uvedenom poradí.

4.3.7 Spúšťanie príkazov na zvolených uzloch

Nebula obsahuje jednoduché prostredie pre spustenie príkazu na vybraných uzloch užívateľovho sliceu. Uvedené rozhranie sa nachádza pod záložkou „Host View“.

Napr., chceme rýchlo skontrolovať aké procesy sú spustené na niekoľkých uzloch PlanetLabu (spustením príkazu „ps“). Postup je nasledovný:

1. Spustíme aplikáciu Nebula a nástroj Plush (menu Plush → Start Plush).
2. Prepne sa do záložky „Host View“, kde sú zobrazené uzly priradené k Vášmu sliceu.
3. Pri stlačenej klávese Ctrl vyberieme servery, na ktorých chceme príkaz spustiť.
4. Do textového okna vedľa tlačidla „Run Command“ zadáme príkaz „ps“.
5. Spustíme príkaz stlačením tlačidla „Run Command“.
6. Stav vykonania príkazu sa zobrazí v stĺpci „Status“.
7. Výstupy sú zobrazené v dolnej časti okna v prostredí „Output“ (Obr. 9.10).

Pozn.: v prípade, že Ste museli pôvodného SSH klienta Mindterm aktualizovať môže sa stať, že výstup príkazov sa zobrazí len v hlavnom okne terminálu, v ktorom bol program Nebula spustený.

4.3.8 Určenie predka procesu

V tejto časti bude uvedený príklad aplikácie, ktorá závisí od poradia spúšťania procesov. Určením predka procesu môžete špecifikovať tok dát v aplikácií.

Napr., chceme stiahnuť domovskú stránku Google a Yahoo a výsledné dáta skomprimovať do tar balíčka. Postup je nasledovný:

1. Spustíme aplikáciu Nebula a nástroj Plush (menu Plush → Start Plush).
2. Prepne sa do záložky „Application View“.

3. Vytvoríme nový komponent (návod časť 4.3.5), ktorý bude obsahovať jeden uzol.
4. Vyberieme vytvorenú položku a kliknutím na tlačidlo „Process“ k nej priradíme proces.
5. Dvojklikom otvoríme editačné okno procesu a vložíme nasledovné údaje.

```
Name: process1  
Path: wget  
Command Args: http://google.com -O google.html
```

Na základe týchto príkazov Plush spustí nástroj wget, ktorý stiahne domovskú stránku Googlu a uloží ju do súboru „google.html“.

6. Vytvoríme ďalší proces, ktorý stiahne domovskú stránku Yahoo a uloží ju do súboru „yahoo.html“.

```
Name: process2  
Path: wget  
Command Args: http://yahoo.com -O yahoo.html
```

7. Na skomprimovanie súborov bude slúžiť tretí proces.

```
Name: process3  
Path: tar  
Command Args: cvf result.tar google.html yahoo.html
```

8. Teraz je potrebné nastaviť poradie procesov tak aby prvé dva skončili pred spustením tretieho. Ak by to bolo opačne tretí proces by nemal pripravené dáta na kompresiu. Správne poradie môžeme nadefinovať určením predchodcov:
 - a) Vyberte prvý proces a pri stlačení Ctrl kliknite na tretí proces. Tým označíte prvý proces za predchodcu tretieho.
 - b) Postup opakujte pre druhý proces. Zobrazenie novovytvoreného komponentu by malo byť zhodné s prílohou (Obr. 9.11).
9. Uložte Plush aplikáciu a spustite ju.

4.3.9 Využívanie softvérových a synchronizačných prvkov

Cieľom tejto časti je vysvetlenie synchronizácie komponentov. Na tento účel sa využívajú tzv. bariérové prvky, ktoré slúžia na synchronizáciu spúšťania procesov vo vnútri distribuovanej aplikácie.

Pre jednoduché vysvetlenie použitia týchto prvkov si predstavte nasledovný prípad. Potrebujete uploadovať súbor s vytvorenou aplikáciou. Táto aplikácia využíva dve skupiny uzlov. Jedna skupina obsahuje servere, na ktorých je súbor s aplikáciou uložený a v druhej skupine sú počítače, ktoré potrebujú tento súbor stiahnuť. Úlohou aplikácie je

experimentovať s využitím distribučnej siete na odoslanie súboru aplikácie zo zdroja k príjemcovi použitím nového druhu protokolu určeného na distribúciu dát. V tomto prípade musí dôjsť k pripojeniu zdrojov a príjemcov k distribučnej sieti pred začiatkom prenosu. Taktiež skupina serverov musí pripraviť súbory na prenos pred tým než si klienti dáta vyžadujú.

Postup je nasledovný:

1. Spustíme aplikáciu Nebula a nástroj Plush (menu Plush → Start Plush).
2. Prepne sa do záložky „Application View“.
3. Kliknutím na tlačidlo „Software“ vytvoríme softvérový komponent.
4. Dvojklikom na vytvorený objekt vyvoláme jeho editačné okno.
5. Do editoru zadáme nasledujúce údaje:

```
Name: server software
Type: tar
Package Type: web
Path: http://147.229.213.xxx/server.tar
Dest Path: server.tar
```

Na základe týchto údajov nástroj Plush vie, že serverovú aplikáciu môže stiahnuť z adresy „http://147.229.213.xxx/server.tar“, a že ju má uložiť pod názvom „server.tar“.

6. Vytvoríme ďalší softvérový komponent, ktorý bude reprezentovať klientskú aplikáciu a do editačného okna vložíme nasledujúce údaje:

```
Name: client software
Type: tar
Package Type: web
Path: http://147.229.213.xxx/client.tar
Dest Path: client.tar
```

7. Klikneme na tlačidlo „Component“ a vytvoríme blok komponentu pre server. V editačnom okne „Add Component Block“ zadáme nasledovné údaje:

```
Name: server
Number of hosts: 1
Type: planetlab
Slice: názov_sliceu
Software: server_software
```

10. Vytvoríme ďalší blok komponentu pre klienta a zadáme nasledujúce údaje:

```
Name: client
Number of hosts: 1
Type: planetlab
Slice: názov_sliceu
Software: client_software
```

11. Vytvoríme príslušné procesy a synchronizačné prvky podľa prílohy (Obr. 9.12).

Poznámka: Všimnite si, že názov synchronizačného prvku je v oboch komponentoch rovnaký. Tým je zaručená synchronizácia procesov medzi serverom a klientom, ktorá zaisťuje, že servery majú pripravené súbory na prenos a spolu s klientmi sú pripojené k distribučnej sieti. Po ukončení synchronizácie uzlov (predpokladáme dostupnosť potrebných súborov) dôjde k spusteniu samotného procesu odosielania a prijímania aplikačných súborov. Za potrebné súbory sú považované súbory „priprav_subory.pl“ a „odosli_subory.pl“ skomprimované do archívu „server.tar“ a súbor „prijat_subory.pl“ v archíve „client.tar“. Súbor „pripoj.pl“ musí byť súčasťou oboch archívov.

4.3.10 Ďalšie funkcie aplikačného modulu

Vývojové prostredie XML aplikácií programu Nebula obsahuje okrem tlačidiel na vytvorenie odpovedajúcich konštrukčných blokov a tlačidla na spustenie aplikácie „Run Application“ aj ďalšie tri funkcie.

Prvou z nich je funkcia „Force Start“, ktorá slúži na spustenie aplikácie v prípade, že ešte neboli pripravené všetky sieťové prostriedky. Táto funkcia je užitočná ak sa Vám nechce čakať na pripojenie ku všetkým uzlom a rozhodnete sa, že ich počet je už dostatočný.

Dve nasledujúce tlačidlá sú užitočné v prípade počiatočného testovania aplikácie. Je to tlačidlo „Abort“ slúžiace na ukončenie vykonávania aplikácie v prípade problému a tlačidlo „Disconnect“ na následné odpojenie od pridelených uzlov.

Výstupy programu Plush pri vykonávaní aplikácie sú uvedené v terminálovom okne, v ktorom bola spustená Nebula alebo je ich možné zobrazit' pomocou položky Plush -> View Plush Output priamo v grafickom rozhraní, vid' Obr. 9.13. Tieto informácie sú vhodné na odhaľovanie chýb pri počiatočnom vývoji aplikácie.

4.3.11 Praktické využitie programu

Ako príklad som použil aplikáciu na lokalizovanie organizácií siete PlanetLab spracovanú v časti venovanej programu CoDeploy. Pre uvedenú aplikáciu je taktiež potrebný web server s podporou PHP. Pri jeho inštalácii a konfigurácii postupujte podľa časti 3.2.4.

Základné požiadavky uvedené v pôvodnom návode v časti 3.3 sú nezmenené. V prípade akýchkoľvek nejasností sa taktiež obráťte na tento návod. Výstup aplikácie bude správny len v prípade, že sa vo Vašom slice bude nachádzať len jeden uzol z každej organizácie. Preto ak už máte vo svojom slice pridané uzly je ich potreba odstrániť, napr. pomocou rozhrania na *stránke*¹ PlanetLabu a následne priradiť nový výber uzlov pomocou postupu v časti 3.3.2. Potrebne dáta na web servery pripravte podľa kapitoly 3.3.3.

Spustenie cron deamona je v uvedenej XML aplikácií riešený pomocou textového dokumentu „locate.cron“, ktorý obsahuje text `*/1 * * * * python phonehome.py` a je ho tiež potreba umiestniť k ostatným súborom na web serveri. Uvedený text špecifikuje program a interval jeho spúšťania, presnejšie má byť spúšťaný uploadovaný skript „phonehome.py“ programom python každú minútu. Na definovanie intervalu slúži päť znakov „*“ a ich význam je nasledovný: [minúty] [hodiny] [deň v mesiaci] [mesiac] [deň v týždni]. Interval spúšťania programu je možné zadať aj viacerými presnými hodnotami. Ak chceme program spúšťať nepravidelne stačí namiesto hviezdičky zadať reťazec čísel oddelených čiarkou, tzn. pre spustenie programu v 5., 10. a 15. minúte v priebehu hodiny je potrebné zadať reťazec `5,10,15` namiesto prvého znaku „*“.

Po príprave všetkých súborov je možné prejsť k vytvoreniu samotnej aplikácie podľa priloženého Obr. 9.14. V prípade nejasností použite návod na obsluhu programu z časti 4.3. Vytvorená aplikácia obsahuje dva softvérové prvky odkazujúce na súbory „phonehome.py“ a „locate.cron“, ktoré je potrebné uploadovať na pridelené uzly. Blok komponentu má pridelených 50 uzlov (nastavte ľubovoľne), využíva parametre SWORD vyhľadávania „resptime“ a „oneminload“ a má priradené dva vytvorené softvérové bloky. Aplikácia je zobrazená pri spustení pričom zelená farba znázorňuje úspešne ukončené procesy. Pravá časť obrázku obsahuje zoznam pridelených zdrojov. Hlavný komponent aplikácie obsahuje sedem procesov s definovaným poradím spustenia, ktoré je treba dodržať. Detailnejší význam procesov je nasledovný:

1. Nastavenie spustiteľnosti skriptu „phonehome.py“.
2. Oznámenie programu a intervalu jeho spúšťania formou súboru „locate.cron“ službe cron.
3. Spustenie deamona služby cron.
4. Pozdržanie vykonávania príkazov, resp. zastavenia služby cron. Čas je pre istotu nastavený tak aby bol skript aplikácie spustený dvakrát.
5. Zastavenie cron deamona.
6. Odstránenie položky z tabuľky služby cron.
7. Odstránenie uploadnutých súborov zo vzdialeného uzlu.

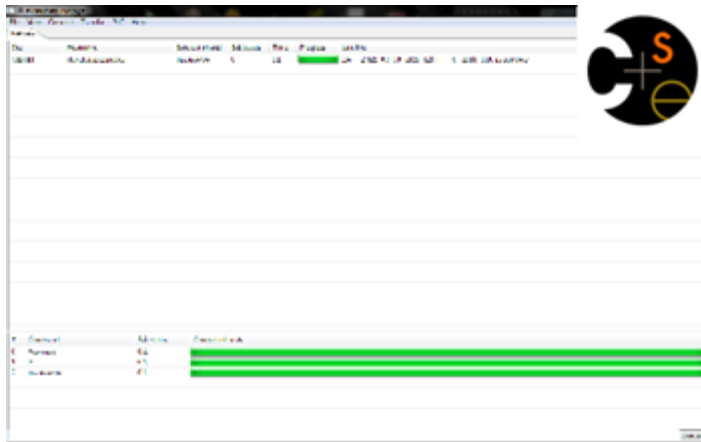
¹ <https://www.planet-lab.org/db/slices/index.php>

Obr. 9.15 v hornej časti obsahuje prostredie zobrazenia uzlov s výberom zobrazovaných stavov procesu spustenia vytvorenej aplikácie na jednotlivých uzloch. V pravej dolnej časti obrázku je znázornené odpovedajúce farebné zobrazenie stavu na mape sveta.

4.3.12 Proces pripojenia k uzlu

Ľavá spodná časť *Obr. 9.15* obsahuje výpis procesov klienta spustených na vzdialenom uzle pri pripojovaní. Plush na začiatku pošle na vzdialený uzol skript „bootstrap.pl“ obsahujúci špecifikáciu procesu pripojenia a ďalšie inštrukcie. Tento skript je následne spustený čím dôjde k stiahnutiu tar balíčku s Plush klientom z definovaného web serveru. Archív je následne rozbalený a dochádza k spusteniu klienta. V tomto bode je programom Plush signalizované úspešné nadviazanie spojenia so vzdialeným uzlom.

5 PLMAN: ALTERNATÍVA PRE WINDOWS



Obr. 5.1: Grafické prostredie programu PIMan.

Alternatívou k programu Plush a jeho grafickej nadstavbe je program PIMan. Potrebne knižnice aj samotný program sú vytvorené v jazyku Java. Z toho vyplýva jeho nenáročnosť a multiplatformnosť. Základné funkcie programu sú takmer totožné s funkciami Plushu. Hlavným rozdielom je možnosť vytvorenia vlastného skriptu využitím definovaných XML-RPC funk-

cií, ktoré sú prístupné aj v grafickom rozhraní programu.

5.1 Popis programu

PlanetLab manažér experimentov (PIMan) zjednodušuje upload, spustenie a monitorovanie vytvorenej aplikácie. Program obsahuje jednoduché grafické rozhranie na vykonávanie bežných úloh ako napr.:

- Priradenie uzlov k sliceu.
- Výber uzlov na základe informácií monitorovacieho systému CoMon.
- Upload experimentu na zvolené uzly.
- Paralelné spúšťanie príkazov na zvolených uzloch.
- Monitorovanie experimentu a zobrazenie konzolového výstupu uzlov, na ktorých je spustený.

Pre skúsenejších užívateľov PIMan obsahuje rozsiahle rozhranie na tvorbu skriptov, ktoré je taktiež súčasťou GUI a môže byť využité na dávkové operácie. Rozhranie využíva XML-RPC komunikáciu čo umožňuje tvorbu skriptov v jazyku Perl, Python, Ruby ako aj v Jave a C/C++. Detailný popis funkcií rozhrania nájdete na *tejto*¹ stránke. Pomocou *tohto*¹ odkazu je taktiež dostupný príklad zložitejšieho Perl skriptu.

¹ http://www.cs.washington.edu/homes/isdal/plman/doc/edu/washington/cs/pl_if/rpc/RpcFunctions.html

5.2 Inštalácia

Z domovskej *stránky*² programu si stiahnite archív s programom pre Váš operačný systém a rozbaľte ho. Balíčky určené pre 32-bitové OS fungujú po dodatočnej konfigurácii, aj v 64-bitových systémoch. V prípade, že nemáte nainštalovanú Javu minimálne vo verzií 1.5 stiahnite si aktuálny inštalačný balíček z *tejto*³ stránky.

V prípade OS Windows 7 Professional x64 je potreba pre správnu funkčnosť Javy v príkazovom riadku pridať do systémovej premennej Path cestu k jej inštalačnému adresáru, pravdepodobne „C:\Program Files (x86)\Java\jre6\bin“. Pred pridaním cesty si overte či už iný program využívajúci Javu nevykonal potrebnú úpravu. Otvorte príkazový riadok a napíšte príkaz „java -version“. Ak sa Vám namiesto výpisu podrobností o verzií aktuálne inštalovanej Javy zobrazí hlásenie, že príkaz nebol rozpoznaný, postupujte nasledovne. Kliknite na Štart → Ovládací panel → Systém → Rozšírené systémové nastavenia. V dolnej časti otvoreného okna kliknite na tlačidlo „Premenné prostredia“, v časti „Systémové premenné“ nájdite premennú Path a stlačte tlačidlo „Upraviť“. Na začiatok pridajte cestu k inštalačnému adresáru Javy vo Vašom systéme a oddel'te ho od nasledujúceho obsahu premennej znakom „;“. Následne potvrd'te zmenu pomocou tlačidla OK vo všetkých troch otvorených oknách. Opätovne overte funkčnosť Javy v novootvorenom okne príkazového riadku zadaním príkazu „java -version“.

V Ubuntu 10.10 x64 je potrebné na *tejto*⁴ stránke stiahnuť odpovedajúci balíček grafických systémových nástrojov „swt-3.6.2-gtk-linux-x86_64.zip“ pre jazyk Java. Z rozbaleného balíčku je potreba skopírovať súbor „swt.jar“ do podadresáru „lib“ v adresári programu PIMan a premenovať ho na „swt-linux_x86_64.jar“. Pri inštalácii programu PIMan použite vždy aktuálnu verziu „swt“ balíčka. V prípade problému môžete skúsiť použiť verziu 3.6.2 priloženú na médiu.

5.3 Konfigurácia a spustenie

V konfiguračnom súbore „pl_manager.conf“ v adresári programu je potrebné uviesť aspoň základné prihlasovacie údaje.

- V položke „username“ uveďte login, pod ktorým sa prihlasujete na stránku PlanetLabu, napr. „user@stud.feec.vutbr.cz“.
- Ďalej uveďte názov sliceu, ktorý Vám bol pridelený, napr. cesnet_s2.

¹ http://www.cs.washington.edu/homes/isdal/plman/Tutorial/advanced_perl_example.pl

² <http://www.cs.washington.edu/research/networking/cplane/>

³ <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

⁴ <http://download.eclipse.org/eclipse/downloads/drops/R-3.6.2-201102101200/index.php#SWT>

- Do položky „PrivateKeyFile“ zadajte úplnú cestu k privátnemu kľúču, ktorý využívate na pripojenie k uzlom PlanetLabu, napr. „/home/user/.ssh/id_rsa“.
- Užívatelia OS Windows môžu na vygenerovanie kľúčového páru použiť program *puttygen*¹. Po spustení programu zvolíte typ kľúču „SSH-2 RSA“ a počet bitov v generovanom kľúči zadajte 2048. Následne stlačte tlačidlo „Generate“ a uveďte passphrase na zabezpečenie privátneho kľúču. Verejný kľúč vytvoríte prekopírovaním vygenerovaného obsahu z hornej časti okna do prázdneho súboru, ktorý pomenujte napr. „id_rsa.pub“. Privátny kľúč vyexportujte pomocou záložky Conversions → Export OpenSSH key do súboru s názvom napr. „id_rsa“. **Pre zadanie cesty k privátnemu kľúču je potreba v systéme Windows využívať dvojité spätné lomítko, napr. „C:\\PlMan\\id_rsa“.**

V konfiguračnom súbore je tiež možné špecifikovať prihlasovacie heslo na stránku PlanetLabu v premennej „AuthString“ a passphrase použitý pri generovaní kľúčov v premennej „PrivateKeyPassword“. **Heslá sú uložené v nešifrovanej podobe!** Z toho dôvodu je ľahšie tieto údaje neuvádzať a vyplniť ich na požiadanie pri každom spustení programu PlMan.

Program je možné vo Windowse spustiť dvojklikom na súbor „run_win_x86.bat“. V linuxe to závisí od konkrétnej platformy, na ktorú ste program nainštalovali. Na spustenie programu v 32-bitovom systéme zadajte príkaz `sh run_linux_x86.sh` v adresári programu, v 64-bitovom je potrebné spustiť skript „run_linux_x86_64.sh“.

Pred spustením GUI je potrebné v príkazovom riadku zadať heslo na stránku PlanetLabu a passphrase. PlMan sa po zadaní požadovaných údajov pripojí k centrálnej databáze PlanetLabu a načíta zoznam uzlov Vášho sliceu.

5.4 Funkcie programu

5.4.1 Výber uzlov na pripojenie

Na lište v záložke „Connect“ máme dostupných hneď niekoľko spôsobov pripojenia. Na pripojenie ku konkrétnemu uzlu po zadaní jeho názvu alebo ip adresy slúži funkcia „Quick connect“. Nasledujúca možnosť „Connect to numbered hosts“ je vhodná na výber počtu uzlov značených podľa vzoru názov uzlu, poradové číslo a doména organizácie/sliceu. Môžeme teda zadať rozsah uzlov, ku ktorým sa chceme pripojiť. Ďalšia možnosť „Connect

¹ <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

to hosts in file“ načíta zoznam uzlov na pripojenie zo súboru s príponou „.hosts“. Ide o textový súbor obsahujúci jeden názov uzlu na riadok. Funkcie „Save connected/finished hosts to file“ uložia zoznam pripojených uzlov resp. tých s ukončeným vykonávaním zadáných príkazov do súboru v rovnakom formáte. Po zvolení možnosti „Select hosts wizard“ sa otvorí okno zobrazené na Obr. 10.1. V ňom si môžete vybrať uzly Vášho sliceu, ku ktorým sa chcete pripojiť na základe príslušnosti k organizácii, názvu a základných CoMon údajov ako je zaťaženie, odozva, voľná operačná pamäť a veľkosť pamäte. Bližšie informácie o uvedených monitorovaných hodnotách nájdete v časti 6.1.5. Ak sa namiesto hodnôt zobrazuje „?“ väčšinou to znamená, že uzol je nefunkčný alebo má blokové porty, ktoré monitorovacia služba využíva. Ďalšou možnosťou výberu je náhodná voľba. Stačí zadať počet uzlov do okna „Quick select“ a potom stlačiť tlačidlo „Random sites“ pre výber uzlu z náhodných organizácií alebo tlačidlo „Random hosts“ pre náhodný výber uzlov.

5.4.2 Spúšťanie príkazov

Po potvrdení výberu uzlov sa program k týmto uzlom pripojí. Pre paralelné vykonanie príkazu na všetkých uzloch zadajte požadovaný príkaz do spodnej časti hlavného okna vyznačenej červeným rámčekom (Obr. 10.2) a spustíte ho tlačidlom „Execute“. V zozname príkazov uvidíte priebeh jeho vykonávania. V prípade, že niektorý z pripojených uzlov na príkaz nereaguje môžete ho zrušiť pomocou RMB a zvolením možnosti „kill command“. V aktuálnej záložke „Overview“ sa zobrazuje zoznam pripojených uzlov s posledným vykonaným príkazom, jeho posledným riadkom výpisu, návratovou hodnotou a čas potrebný na jeho dokončenie resp. čas, po ktorom bol zrušený. Po dvojkliku na názov uzlu sa otvorí záložka s jeho terminálovým výstupom. V tomto rozhraní je taktiež uvedený zoznam vykonaných príkazov spolu s ich návratovou hodnotou a potrebným časom ale nachádza sa tu aj ich kompletný terminálový výstup.

5.4.3 Pridanie uzlov k sliceu

Uzly k sliceu je možné priradiť pomocou položky „Add Hosts to Slice“ z menu „PLC“ na hornej lište. Zobrazí sa rovnaké okno ako bolo popísané v kapitole 5.4.1 s tým, že jeho obsahom je zoznam uzlov, ktoré nie sú súčasťou Vášho sliceu. Po potvrdení výberu môže trvať aj niekoľko hodín kým dôjde k vytvoreniu Vášho sliceu na všetkých serveroch a k rozšíreniu Vášho verejného kľúča.

Druhou možnosťou je pridanie uzlov zo súboru „Add Hosts in File to Slice“. Názvy uzlov musia byť uložené v textovom dokumente s príponou „.hosts“ s maximálne jedným názvom uzlu na riadok.

5.4.4 Upload súborov

PIMan umožňuje spoľahlivý upload súborov na pripojené uzly pomocou funkcií „Upload Directory/File“ v menu „Transfer“ na hornej lište. Pri uploade celého adresáru sú rekurzívne skopírované aj všetky jeho podadresáre do adresáru s rovnakým názvom ako je názov uploadovaného adresáru. Tento adresár je vytvorený na vzdialených uzloch v domovskom adresári sliceu. V prípade uploadu jedného súboru je tento súbor taktiež skopírovaný do domovského adresáru. Proces uploadu pozostáva zo štyroch krokov:

1. Výpočet kontrolného md5 súčtu pre lokálny súbor/y.
2. V prípade, že súbor/y na vzdialenom uzle existuje je vypočítaný aj jeho kontrolný md5 súčet.
3. Upload súboru nasleduje v týchto dvoch prípadoch: súbor na vzdialenom uzle neexistuje alebo sa jeho kontrolný súčet nezhoduje s kontrolným súčtom lokálneho súboru.
4. Po ukončení prenosu sa znovu prekontroluje či navzájom súhlasia kontrolné súčty lokálneho a vzdialeného súboru. Tým je overená bezchybnosť prenosu.

5.4.5 Download súborov

Súbory zo vzdialených uzlov je možné stiahnuť pomocou nástroja „Download Wizard“ v menu „Transfer“ na hornej lište. Obr. 10.3 znázorňuje okno s nastavením sťahovania a postup tohto procesu zobrazený v terminálovom výpise daného uzlu. Položka „Local target directory“ slúži na nastavenie lokálneho adresáru, do ktorého sa súbor alebo adresár stiahne. „Remote directory or file“ je názov adresáru alebo súboru na stiahnutie. Nasledujúce textové okno slúži na zadanie vzoru porovnávacieho grep filtru, ktorý je aplikovaný na zadanú vzdialenú cestu a tým umožňuje výber požadovaných súborov podľa ich názvu. Ak nechcete zadať žiadny filter ponechajte položku prázdnu. Posledné nastavenie „name convention“ slúži na výber medzi týmito dvoma spôsobmi lokálneho pomenovania súborov:

Flat – všetky súbory budú skopírované do toho istého lokálneho adresáru. Na pomenovanie súborov sa využíva nasledujúci vzor. V prípade, že chcete stiahnuť súbor „/tmp/output.log“ zo vzdialeného uzlu „planetlab2.cesnet.cz“, súbor bude lokálne pomenovaný nasledovne:

```
lokálny_adresár/planetlab2.cesnet.cz_tmp_output.log
```

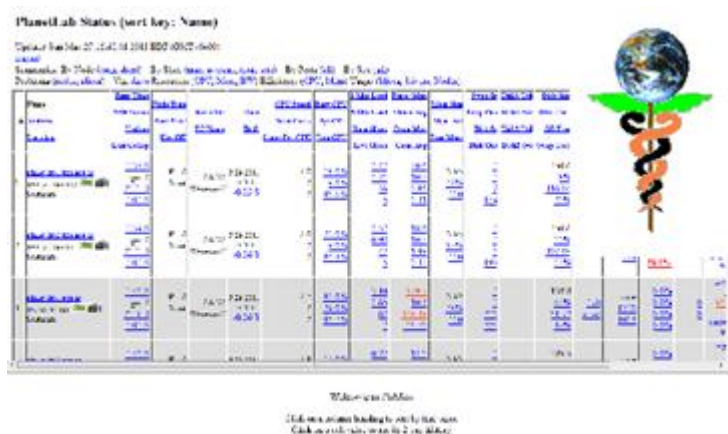
Hierarchical – v lokálnom adresári bude vytvorená odpovedajúca adresárová štruktúra. V prípade rovnakého príkladu ako v predchádzajúcom bode bude súbor uložený nasledovne:

```
lokálny_adresár/planetlab2.cesnet.cz/tmp/output.log
```

Postup pri sťahovaní súboru je rovnaký ako pri uploadovaní s tým rozdielom, že sa jedná o prenos zo vzdialeného uzlu do lokálneho adresáru.

6 MONITORING

6.1 CoMon



Obr. 6.1: HTML prostredie služby CoMon.

CoMon poskytuje rozsiahlu monitorovaciu štatistiku pre PlanetLab na úrovni uzlov a sliceov. Môže byť použitý napríklad na odhalenie nepriaznivých vplyvov na výkon uzlov alebo na kontrolu priradených prostriedkov pre jednotlivé experimenty.

Táto služba bola od svojho vzniku voľne prístupná v celej sieti PlanetLabu. Z rastom siete sa ale zvyšovalo aj jej zaťa-

ženie a preto sa tvorcovia rozhodli prístup obmedziť.

Výskumná sieť CESNET, pod ktorú spadá aj sieť FEKT VUT poskytuje svoje uzly na voľné využitie v rámci siete PlanetLab a preto má právo na bezplatné využitie tejto služby. Preto ak sa nachádzate v adresnom priestore 147.229.0.0 – 147.229.255.255 bude Vám prístup povolený. Ak chcete túto službu využívať v inej sieti informujte sa u správcu alebo Vášho vedúceho či spĺňate podmienku bezplatného alebo plateného prístupu a následne zažiadajte o jeho pridelenie.

6.1.1 Prehľad

Spôsob využitia služby CoMon závisí od toho čo od nej vyžadujete. V prípade podozrenia, že jeden z experimentov sa chová na niektorých uzloch podozrivo je užitočné použiť zobrazenie na úrovni uzlov. To umožní porovnanie štatistiky uzlov medzi sebou a jednoduché odhalenie parametru mimo opodstatnenej hranice. Ďalším príkladom využitia môže byť zistenie množstva použitej pamäte v prípade dlhodobého testovania. Jej aktuálne vyťaženie môžete sledovať ako celok v rámci celého sliceu alebo si môžete zobrazit spotrebu jednotlivých uzlov. Napr., po zobrazení stránky „Slice Max“ môžete zistiť maximum využitej pamäte na jednotlivých uzloch a následným kliknutím na túto hodnotu zobrazit graf dvoj dňovej histórie. Podobným spôsobom sa dajú odhaliť rôzne problémy, ktoré zbytočne spotrebúvajú zdroje sliceu alebo konkrétneho uzlu.

Prakticky všetko v monitorovacom systéme CoMon je zoraditeľné pomocou kliknutia na názvy stĺpcu. V zobrazení na úrovni uzlov sú vo väčšine uzlov zobrazené dve hodnoty a preto má aj daný stĺpec dva rozkliknuteľné názvy.

Takmer každá hodnota je tiež graficky zobraziteľná. Kliknutie na ktorúkoľvek hodnotu zobrazí graf dvoj dňovej histórie zvoleného parametru. Hodnoty dostupných grafov nie sú zobrazované v reálnom čase z dôvodu generovania väčšieho počtu dát ako dokáže disk monitorovacej služby spracovať. Nadmerné dáta samozrejme nie sú zahadzované, len spracované s určitým malým oneskorením.

6.1.2 Formát požiadaviek a výstupných dát

CoMon podporuje výber jednotlivých riadkov štatistiky na základe užívateľských kritérií. Požiadavky na monitorovanie podporujú výrazy jazyku C, na základe ktorých sú zobrazené len riadky spĺňajúce zadané podmienky. Aktuálna sada podporovaných operátorov zahŕňa operátory porovnávania (>, <, >=, <=, ==, !=), zátvorky a znaky logického a/alebo (&&, ||). Názvy stĺpcov sú reprezentované alfanumerickými znakmi bez medzier a podporované sú aj konštanty.

Pre jednoduchšie použitie dát služby CoMon pre rôzne druhy skriptov je možné špecifikovať formát výstupných dát. Defaultným zobrazením výstupu sú html tabuľky v užívateľsky prívetivom formáte. Možnosti pre zobrazenie dát pre skripty sú „nameonly“ pre zobrazenie iba názvov riadkov, „formatcsv“ pre hodnoty oddelené čiarkou a „formatspaces“ pre hodnoty oddelené medzerou. Je možné zadať aj limit pre zobrazený počet riadkov a zoradiť ich podľa zvoleného stĺpcu.

6.1.3 Vzorové požiadavky

Nasleduje zoznam vzorových požiadaviek využívajúcich všetky dostupné možnosti zobrazenia výstupných dát. Pre zobrazenie podľa vybraného parametru pridajte jeho formálny zápis do *tohto*¹ základného CGI skriptu programu CoMon.

Aktívne uzly – uzol reaguje na požiadavky služby CoMon. Používa premennú
`select='resptime > 0'.`

Často využívané uzly – viac ako 50 sliceov uložených v pamäti, z ktorých je aspoň 10 aktívnych. Používa: `select='liveslices >= 10 && numslices > 50'.`

Mierne zaťažené uzly – vyhľadáva uzly, ktoré odpovedajú (nie sú nefunkčné), majú zaťaženie po dobu jednej minúty menšej ako 5 (počet aktívnych procesov) a obsahujú 5 a menej sliceov. Používa: `select='resptime > 0 && lminload < 5 && liveslices <= 5'.`

¹ http://comon.cs.princeton.edu/status/tabulator.cgi?table=table_nodeviewshort&

Nestále uzly – odchýlka času väčšia ako jedna minúta. Používa: `select='drift > 1m'`.

Pretiažený primárny DNS – sekundárny DNS je menej zaťažený ako primárny. Používa: `select='dnsludp >1 && dns2udp >= 0 && dnsludp > dns2udp'`.

Závažné problémy s DNS – primárny aj sekundárny DNS sú nefunkčné. Používa: `select='dnsludp > 80 && dns2udp > 80'`.

Málo miesta na disku – veľkosť voľného diskového priestoru menšia ako 5 GB. Používa: `select='resptime > 0 && gbfree < 5'`.

Zlyháva SSH – na uzol sa nebolo možné prihlásiť prostredníctvom SSH po dobu dlhšiu ako dve hodiny. Používa: `select='sshstatus > 2h'`.

Viacero problémov – zlúči zoznamy uzlov s veľkou odchýlkou času, DNS problémami, voľným diskovým priestorom pod 5 GB a SSH problémami. Používa: `select='drift > 1m || (dnsludp > 80 && dns2udp > 80) || (resptime > 0 && gbfree < 5) || sshstatus > 2h'`.

Aktívne a bezproblémové uzly – odstráni zo zoznamu aktívnych uzlov tie, ktoré majú niektorý z predchádzajúcich problémov. Používa: `select='resptime > 0 && ((drift > 1m || (dnsludp > 80 && dns2udp > 80) || gbfree < 5 || sshstatus > 2h) == 0)'`.

Málo zaťažené bezproblémové uzly – odstráni zo zoznamu mierne zaťažených uzlov tie, ktoré majú niektorý z predchádzajúcich problémov. Používa: `select='(resptime > 0 && 1minload < 5 && liveslices <= 5) && ((drift > 1m || (dnsludp > 80 && dns2udp > 80) || gbfree < 5 || sshstatus > 2h) == 0)'`.

Triedenie podľa stĺpca – výsledky môžu byť zoradené zadaním názvy zvoleného stĺpca. Napr. môžeme použiť zobrazenie podľa záťaže počas posledných piatich minút `sort=5minload`. Poradie zoradenia môžeme invertovať zadaním negatívneho názvu stĺpca `sort=-5minload`.

Počet vybraných uzlov organizácie – počet zobrazovaných uzlov jednej organizácie môžeme obmedziť na najlepšie uzly. Na zobrazenie jedného uzlu na organizáciu je potrebné pridať nasledujúcu premennú: `persite=1`. Počet zobrazených uzlov môže byť ľubovoľne špecifikovaný.

Výber počtu zobrazených riadkov – počet riadkov v zobrazenom výstupe sa dá obmedziť nasledovne: `limit=10`.

Výber formátu a jednotlivých stĺpcov – na zjednodušenie exportu dát a následnej analýzy je možné zvoliť namiesto HTML zobrazenia formát CSV a vybrať len požadované stĺpce. Pre voľbu formátu CSV zadajte `format=formatcsv` a pre

následné vypustenie napr. stĺpcov názvu, umiestnenia a doby chodu uzlu zadajte `dumpcols='name,location,uptime'`.

Zobrazenie názvov – pre zjednodušenie skriptovania je lepšie použiť zobrazenie samostatných názvov uzlov namiesto plnohodnotnej tabuľky pomocou premennej `format=nameonly`.

6.1.4 Prístup k dátam

Najlepší spôsob prístupu k dátam je cez už uvedené rozhranie akceptujúce jednotlivé požiadavky. Názvy stĺpcov a formát zápisu jednotlivých požiadaviek je takmer nemenný. Ak ale potrebujete prístup k neupraveným dátam priamo z uzlov je možné použiť druhého alternatívneho daemona, ktorý je spustený na každom uzle. Daemon na úrovni uzlov je prístupný cez port 3121 a po uskutočnení spojenia ihneď zasiela dáta. Nie je potrebné špecifikovať žiadne požiadavky. Daemon na úrovni sliceov načúva na porte 3120 a reaguje na http požiadavku „/cotop“. Požiadavka na slice dáta napr. pre uzol „planetlab2.cesnet.cz“ vyzerá *takto*¹.

Práve dáta z týchto senzorov sú využívané na generovanie výstupných dát služby CoMon. Tieto hodnoty sú taktiež archivované do „monall“ (úroveň uzlov) a „topall“ (úroveň sliceov) záloh. Prvý riadok zálohy „monall“ obsahuje čas vytvorenia zálohy. Dáta každého uzlu sú zapísané do riadkov a oddelené medzerou. Riadok odpovedajúci konkrétnemu uzlu obsahuje jeho názov a od ďalšieho záznamu je oddelený prázdny riadok. Formát zálohy „topall“ má skôr štýl web zobrazenia. Dáta každého uzlu sú oddelené prázdny riadok a začínajú riadkom so slovom „Start“ a údajom času záznamu. Nasleduje názov uzlu a výpis dát z daemona.

Archivované dáta obsahujú výsledky pravidelných päť minútových kontrol. Sú dostupné cez tieto odkazy:

```
„topall“ - http://comon.cs.princeton.edu/status/dump_cotop_YYYYMMDD.bz2  
„monall“ - http://comon.cs.princeton.edu/status/dump_comon_YYYYMMDD.bz2
```

Reťazec „YYYYMMDD“ samozrejme nahraďte rokom, mesiacom a dňom zálohy požadovaných dát. Záznam aktuálneho dňa nemá príponu „bz2“ a nie je komprimovaný. Staršie dáta môžu byť presunuté do iného adresáru alebo offline. V prípade potreby kontaktujte kolektív vývojárov *mailom*².

¹ <http://planetlab2.cesnet.cz:3120/cotop>

² princeton_comon@lices.planet-lab.org

6.1.5 Popis prvkov html rozhrania

V tejto časti je uvedený popis jednotlivých prvkov hlavnej stránky monitorovacieho systému CoMon. Kliknutím na názov stĺpcu sa zoznam zoradí podľa zvolenej hodnoty. Po kliknutí na konkrétnu hodnotu sa zobrazí graf jej dvojdňovej histórie. Html rozhranie je dostupné cez *tento*¹ odkaz.

Popis jednotlivých volieb zobrazenia

Výpisy všetkých volieb sú obmedzené na 50 položiek. Na konci týchto skrátených zoznamov je vždy zobrazené tlačílo s dvoma šípkami na zobrazenie kompletného zoznamu.

Zobrazenie podľa uzlu (By Node) – zobrazenie má dva režimy, režim „long“ a „short“. Zvolením možnosti „long“ sa zobrazia všetky dostupné dáta o uzle. Možnosť „short“ zobrazí najčastejšie sledované hodnoty.

Podľa sliceu (By slice) – zobrazenie má štyri režimy: max, average, total, site. Každý zobrazuje inú matematicky určenú hodnotu z priebehu jednotlivých parametrov sledovaných sliceov. Režim „max“ zobrazuje maximálnu hodnotu parametrov za dvojdňovú sledovaciu dobu, režim „average“ priemernú hodnotu zaokrúhlenú na dve desatinné miesta, režim „total“ celkovú sumu nameraných hodnôt z jednotlivých meracích intervalov a režim „site“ zobrazuje celkovú sumu nameraných hodnôt režimu „total“ na sliceoch v rámci jednej organizácie.

Podľa portov (By Ports) – obsahuje režim „all“, ktorý zobrazuje štatistiku využitia portov na danom uzle pre porty typu Raw, ICMP, porty dlhodobo využívané a zobrazuje počet aktuálne využívaných portov.

Podľa organizácie (By Site) – obsahuje režim „all“, ktorý zobrazí výpis štatistiky jednotlivých organizácií.

Zobrazenie problémov (Problems) – pozostáva z dvoch režimov: nodes a slices. Režim „nodes“ zobrazuje zoznamy po 50 uzloch roztriedené podľa konkrétneho problému. Uzly sú zoradené zostupne (od najhoršieho). Zobrazujú sa zoznamy uzlov s týmito problémami: nedostupné pre SSH prihlásenie po dobu dlhšiu ako jedna hodina, SSH kľúč uzlu sa nezhoduje s kľúčom zaznamenaným v centrálnej databáze, súborový systém neprístupný pre zápis viac ako jednu hodinu, počet sliceov je nulový – indikuje zlyhanie modulu CoStat, SSH prihlásenie funguje ale služba CoMon neobdržala žiadne dáta – pravdepodobne

¹ <http://comon.cs.princeton.edu/status/>

filtrovaný port na prenos dát, preťaženie DNS serverov, časová odchýlka väčšia ako jedna minúta, menej ako 5 GB voľného diskového priestoru.

Režim „slices“ obsahuje uzly s hodnotou niektorého sledovaného parametru nad bežnú úroveň. Uzly sú roztriedené do zoznamov podľa sliceov, ktoré daný problém spôsobujú. Nad každou tabuľkou je umiestnený názov sliceu s krátkou štatistikou a počtom smerodajných odchýlok od priemeru.

Vizualizácie (Viz) – popisu aplikácie, ktorá vizualizácie zostavuje ako aj popisu jednotlivých zobrazovaných hodnôt sa venuje časť 6.2.

Popis názvov jednotlivých stĺpcov a odpovedajúcich hodnôt

Údaje s významom maxima, priemeru a sumy sú vypočítavané z dvojdnového záznamu danej veličiny.

Node Name (Name) – názov uzlu

Address – IP adresa uzlu

Názvy uzlov sú zoradené abecedne. Po kliknutí na názov uzlu sa zobrazia detaily o aktivite na tomto uzle. Pomocou ikony fotoaparátu je možné zobrazit' posledný záznam údajov služby CoTop. Ikona vľavo slúži na zobrazenie aktuálnych CoTop dát, ktoré sú aktualizované každých 5 sekúnd.

Location	– umiestnenie organizácie, pod ktorú uzol patrí
Resp Time	– čas, za ktorý uzol odpovedá na požiadavky
SSH Status	– stav SSH prihlasovania na základe reakčného času uzlu
Uptime	– doba nepretržitej prevádzky uzlu
Last CoTop	– čas od zaslania posledných dát pre monitorovaciu službu
Node Type	– typ uzlu
Boot State	– stav, v ktorom sa systém uzlu nachádza
Key OK	– kontrola zhody kľúča uzlu s kľúčom v centrálnej databáze
Kern Ver	– aktuálna verzia kernelu
FC Name	– kódové označenie verzie kernelu
Date	– dátum nastavený na uzle
Drift	– oneskorenie nastaveného času na uzle oproti centrálnemu
CPU Speed	– takt procesora uzlu v GHz
Num Cores	– počet procesorových jadier
Cores Per CPU	– počet jadier na jeden procesor
Busy CPU	– percentuálne využitie procesorového času
Sys CPU	– percentuálne využitie procesorového času systémom
Free CPU	– nevyužitý procesorový čas v percentách
1 Min Load	– priemerný počet aktívnych procesov počas jednej minúty
5 Min Load	– priemerný počet aktívnych procesov počas piatich minút
Num Slices	– počet vytvorených sliceov

Live Slices	– počet aktívnych sliceov za sledované obdobie, zobrazené sú tie, ktoré využili aspoň desatinu percenta procesorového času za posledných 5 minút
Timer Max	– maximálny interval medzi dvoma časovačmi
Timer Avg	– priemerná hodnota intervalu medzi časovačmi

Na každom uzle je spustený časovač, ktorý sa snaží prejsť do režimu spánku na 1 ms počas jednej sekundy. Predchádzajúce dve hodnoty zobrazujú priemer a maximum intervalu medzi jednotlivými časovačmi po dobu 5 minút a tým informujú o presnosti uzlu pri plánovaní úloh. Zobrazené hodnoty sú v milisekundách.

Conn Max	– maximálny čas potrebný na vytvorenie spojenia v ms
Conn Avg	– priemerný čas potrebný na vytvorenie spojenia v ms
Mem Size	– veľkosť inštalovanej pamäte v GB
Mem Act	– časť operačnej pamäte aktívne využívaná systémom
Free Mem	– výstup testovacieho programu, ktorý vráti hodnotu veľkosti pridelennej pamäte z požadovaných 100 MB, hodnoty nad 90 MB znamenajú relatívne malý nátlak na prerozdelenie pamäte
Swap In	– priemer zapísaných dát do swapovacieho priestoru v MB
Swap Out	– priemer načítaných dát zo swapovacieho priestoru v MB
Disk In	– priemerný objem zapísaných dát na disk v MB
Disk Out	– priemerný objem dát načítaných z disku v MB
Disk1 Util	– priemerné percentuálne vyťaženie prvého disku
Disk2 Util	– priemerné percentuálne vyťaženie druhého disku
Disk Size	– veľkosť inštalovaného diskového priestoru
Disk Used	– priemerné využitie diskového priestoru v percentách
GB Free	– priemerná veľkosť voľného diskového priestoru v GB
Swap Used	– percentuálne využitie swapovacieho priestoru na disku
FD Test	– test dostupných popisovačov súborov, ak je zobrazená iná hodnota ako 0x0 to znamená, že v sledovanom období uzol trpel nedostatkom popisovačov
File RW	– v prípade chyby súborového systému, následného prechodu do režimu iba na čítanie a pretrvávajú problému po dobu dlhšiu ako jedna hodina je tento stav indikovaný uvedenou premennou
BW Limit	– limit šírky prenosového pásma v kb
Tx Rate	– priemerný objem odoslaných dát v kbps
Rx Rate	– priemerný objem prijatých dát v kbps
DNS1 udp	– priemerná chybovosť DNS1 udp požiadaviek
DNS1 tcp	– priemerná chybovosť DNS1 tcp požiadaviek
DNS2 udp	– priemerná chybovosť DNS2 udp požiadaviek

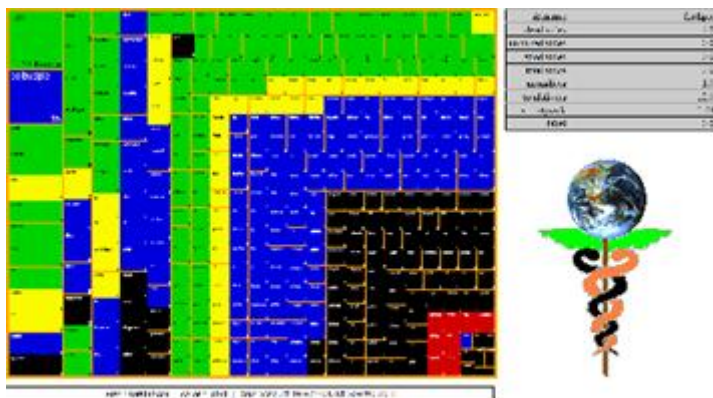
DNS2 tcp	– priemerná chybovosť DNS2 tcp požiadaviek
Raw Ports	– počet využívaných Raw portov na uzle
ICMP Ports	– počet využívaných ICMP portov na uzle
Long Ports	– počet portov využívaných rovnakým procesom/pripojením za poslednú hodinu
Snap Ports	– počet aktuálne využitých portov
Non 206	– kontrola bezchybnosti prenosu dát z uzlu na webserver
Has Via	– zobrazuje identifikáciu medziľahlého zariadenia v prípade proxy serveru
CPU Hog	– zobrazuje názov sliceu najviac zaťažujúci CPU
CPU %	– percentuálna hodnota zaťaženia CPU uvedeným sliceom, v prípade viac jadrového CPU hodnota aj nad 100%
Mem Hog	– zobrazuje názov sliceu najviac využívajúceho pamäť
Mem %	– percentuálne využitie pamäte týmto sliceom
Proc Hog	– zobrazuje názov sliceu, pod ktorým je vytvorených najviac procesov
Num Proc	– počet spustených procesov pre uvedený slice
Tx Hog	– zobrazuje názov sliceu, ktorý odoslal najviac dát
Tx Kb	– 15-minútový priemer odoslaných dát pre uvedený slice
Rx Hog	– zobrazuje názov sliceu, ktorý prijal najviac dát
Rx Kb	– 15-minútový priemer prijatých dát pre uvedený slice
1-min Transmit	– priemerné množstvo odoslaných dát za jednu minútu
15-min Transmit	– priemerné množstvo odoslaných dát za 15 minút
1-min Recieve	– priemerné množstvo prijatých dát za jednu minútu
15-min Recieve	– priemerné množstvo prijatých dát za 15 minút

Priemerné hodnoty odoslaných a prijatých dát sú uvedené v kbps.

Slice Name	– názov sliceu
Phys Mem MB	– priemerné využitie operačnej pamäte v MB
Virt Mem MB	– priemerné využitie virtuálnej pamäte v MB
# Nodes	– celkový počet uzlov priradených k danému sliceu
Site Name	– zjednodušený názov organizácie
# Slivers	– počet vytvorených virtuálnych serverov
Dead Nodes	– počet uzlov organizácie mimo prevádzku
Impaired Nodes	– počet problémových uzlov
Good Nodes	– počet funkčných uzlov
Total Nodes	– celkový počet registrovaných uzlov
Total Slivers	– celkový počet virtuálnych serverov vytvorených na uzloch organizácie
Activity Rank	– hodnotenie aktivity organizácie
Label	– označenie/typ organizácie, používané sú označenia: donor – poskytovateľ, active/inactive – aktívna/neaktívna, trying –

nováčik, leech – označenie v prípade nulového počtu funkčných uzlov, tzv. „príživník“

6.2 CoVisualize



Obr. 6.1: Prostredie vizualizácií.

Služba CoVisualize poskytuje niekoľko vizualizácií využitia siete PlanetLab, ktoré zobrazujú rôzne namerané hodnoty a sú aktualizované každých 5 minút. Potrebne dáta sú poskytované projektom CoMon.

Grafické zobrazenie služby je navrhnuté tak aby uľahčovalo administráciu PlanetLabu. Poskytuje základné informácie, ktoré umožňujú rýchle odhalenie

případných experimentov s pozoruhodným správaním. Taktiež poskytuje proporčné zobrazenie využitia dostupných prostriedkov siete, ktoré je z dát poskytovaných službou CoMon len ťažko zistiteľné. Celkovo je služba CoVisualize vytvorená tak aby poskytovala jednoduchú, užitočnú a graficky dobre spracované zobrazenie aktivity siete.

Pre rýchly prehľad siete si môžete pozrieť automaticky sa obnovujúcu *prezentáciu*¹. Odkaz je dostupný aj v html rozhraní CoMon s časti volieb vizualizácií „Viz“, voľba „Auto“.

6.2.1 Vizualizácie

Aplikácia CoVisualize aktuálne poskytuje osem druhov zobrazení v troch kategóriách: prostriedky, efektivita a využitie. Kategória prostriedkov je zameraná na využitie prenosového pásma, CPU a pamäte jednotlivými sliceami. Zobrazenie efektivity znázorňuje ako slice zaťažuje CPU a pamäť v pomere k množstvu odosielaných a prijímaných dát. Posledné znázornenie slúži na zobrazenie využitia siete PlanetLab jednotlivými sliceami, v zmysle počtu experimentov, uzlov a sliverov.

Každý štvorec grafického zobrazenia obsahuje názov a dve hodnoty. Názov zodpovedá sledovanému sliceu alebo organizácii a uvedené hodnoty zodpovedajú dátam, z ktorých je vizualizácia vytvorená. Prvá hodnota určuje veľkosť štvorca a druhá jeho farbu. Popis

¹ <http://comon.cs.princeton.edu/status/slideshow.cgi>

jednotlivých zobrazení je uvedený v nasledujúcej časti. Základným pravidlom ale je, že červená farba zobrazuje prekročenie bežnej hranice, čierna priemer a zelená žiadanú úroveň sledovanej hodnoty.

V zátvorke za názvom skupiny vizualizácií alebo hodnoty daného typu zobrazenia bude uvedená skratka, pomocou ktorej si môžete vizualizáciu zobrazit' v html rozhraní služby CoMon.

Prostriedky (Resources)

Zobrazenia znázorňujú využitie troch hlavných prostriedkov jednotlivými sliceami.

Procesor (CPU) – zobrazuje využitie procesoru každým aktívnym sliceom PlanetLabu. Veľkosť štvorca znázorňuje podiel využitia procesorového výkonu dostupného v sieti. Slice, ktoré spotrebúvajú pod 0,2% výkonu sú zahrnuté pod jeden štvorec s názvom „Others“, čiže ostatný. Zobrazená hodnota určujúca podiel využitia vyjadruje počet využívaných procesorov v rámci celého sliceu. Ak slice napr. obsahuje sedem uzlov a využíva 30% výkonu z každého, znamená to, že potrebuje celkovo 2,1 procesoru. Farba zobrazuje koľko je určené využitie nad (červená) alebo pod (zelená) úrovňou týždňového priemeru sliceu. Na jej určenie sa využíva vypočítaná smerodajná odchýlka stanovená za posledný týždeň. Dve smerodajné odchýlky sú zobrazené 50% kombináciou červenej a zelenej, zatiaľ čo počet štyroch a viac odpovedá plnej červenej alebo zelenej. Štvorcu s názvom „Others“ nie je priradená žiadna farba pretože priemerná hodnota využitia pre zlúčené slice nie je sledovaná.

Pamäť (Mem) – zobrazuje celkové využitie pamäte sliceu. Hodnotou je počet GB a farbu štvorca určuje počet smerodajných odchýlok od týždňovej priemernej hodnoty sliceu. Ako u CPU, dve sm. odchýlky znamenajú 50% červenej a zelenej a štyri a viac plnohodnotnú farbu. Tí čo spotrebúvajú málo sú opäť zahrnutí do štvorca „Others“. V podstate slice princeton_comon bude často najväčším spotrebiteľom pamäte z dôvodu neustáleho testovania. Tento test určuje okrem iného aj veľkosť pamäte, ktorá môže byť sliceu pridelená. Ak je teda úroveň využitia pamäte v rámci siete PlanetLab malá, test pamäte na slici princeton_comon dostane dodatočné prostriedky.

Prenosové pásmo (BW) – zobrazuje celkový objem prijatých a odoslaných dát v rámci sliceu. Zobrazená hodnota je 15-minútovým priemerom v jednotkách Mbps. Metóda určenia farby je rovnaká ako u CPU a pamäte. Veľkosť štvorca avšak zodpovedá druhej mocnine využitého prenosového pásma namiesto lineárnej závislosti. Dôvodom je omnoho väčšie kolísanie hodnôt využitia prenosového pásma ako u CPU a pamäte. Ak by bola veľkosť štvorca určovaná lineárne tak by došlo k zobrazeniu len veľmi malého množstva sliceov.

Efektivita (Efficiency)

Sieť PlanetLabu je navrhnutá pre sieťový výskum (nie len na výpočty) a uzly potrebujú medzi sebou komunikovať. Uvedené metódy vizualizácie sú preto zamerané na zobrazenie využitia prenosového pásma v závislosti od využitia CPU a pamäte.

Procesor vs prenosové pásmo (CPU) – zobrazuje využitie CPU rovnakým spôsobom ako zobrazenie využitia prostriedkov CPU s veľkosťou štvorca reprezentujúcou časť využitia CPU prostriedkov PlanetLabu. Farba je ale funkciou využitia prenosového pásma s najvyšším využitím zobrazeným plnou zelenou a najnižším plnou červenou. Keďže využitie prenosového pásma má široký rozptyl, farby sú priradené jeho logaritmu. Farba štvorca s ostatnými sliceami je určená zo sumy nimi využívaného prenosového pásma. Všeobecne sú za nežiaduce považované veľké červené štvorce pretože indikujú veľké využitie CPU ale žiadnu komunikáciu. Takéto slice môžu napr. testovať nové projekty alebo v horšom prípade sú zasiahnuté uzavretou slučkou v sieti.

Pamäť vs prenosové pásmo (Mem) – zobrazuje využitie pamäte v závislosti od využitia prenosového pásma. Používa rovnakú metódu určenia farby ako predošlé porovnanie. Veľké červené štvorce sú opäť nežiaduce z dôvodu veľkého využitia pamäte ale žiadnej komunikácie. Tento prípad je ale ťažšie odstrániteľný ako pri nadmernom využití CPU pretože spotreba pamäte sa dynamicky nemení v závislosti od záťaže.

Využitie (Usage)

Posledná skupina vizualizácií ilustruje využitie PlanetLabu na úrovni organizácií. Keďže jednotlivé organizácie poskytujú pre využitie v sieti vlastné uzly naskytá sa tým možnosť sledovať spôsob použitia týchto zdrojov.

Slice (Slices) – poskytuje zobrazenie celkového počtu aktívnych sliceov v sieti PlanetLabu zoskupených podľa organizácie. Veľkosť štvorca znázorňuje počet sliceov v organizácii a farba je založená na hodnotení stavu organizácie. Zelená znamená, že organizácia má aspoň dva funkčné uzly zatiaľ čo žltá znázorňuje, že poskytované uzly majú niektorý z uvedených problémov (nedostatok voľného miesta na disku, veľké časové oneskorenie, DNS problémy ...). Červený štvorec zobrazuje tzv. „leecherov“. Sú to organizácie, ktoré aktívne využívajú PlanetLab ale neposkytujú žiadne funkčné stroje.

Virtuálne servery (Slivers) – zobrazenie je podobné predošlému ale zobrazuje celkový počet sliverov v rámci organizácie. Čím väčší počet serverov je zahrnutých v danom slici tým väčší počet sliverov. Farebná schéma sa zhoduje zo zobrazením sliceov. Počet virtuálnych serverov má minimálny vplyv na zaťaženie. Zobrazenie prostriedkov ako sú CPU, pamäť a prenosové pásmo sú lepším indikátorom využitia.

Uzly (Nodes) – zobrazuje počet registrovaných uzlov organizácie v PlanetLab databáze. Čím viac uzlov organizácia zahŕňa tým je zobrazený štvorec väčší. Počet farieb štvorcov je vyšší. Zobrazenia počtu sliceov a virtuálnych serverov totižto zobrazujú len aktívne slice. V prípade uzlov sú potrebné dve ďalšie kategórie. Prvá znázorňuje skutočnosť, že organizácia poskytuje funkčné uzly ale prostriedky siete momentálne nevyužíva. Tieto organizácie sa nazývajú poskytovatelia (donors) a sú zobrazené modrou farbou. Druhým prípadom je, keď organizácia obsahuje len nefunkčné uzly a PlanetLab nevyužíva. Organizáciu nazývame neaktívna (inactive) a je zobrazená čiernou farbou.

Zhrnutie významu jednotlivých farebných zobrazení:

- Zelená* – organizácia je v dobrom stave
- Žltá* – org. obsahuje len problémové uzly
- Modrá* – org. má funkčné uzly ale žiadnu aktivitu
- Čierna* – org. nemá funkčné uzly a je neaktívna
- Červená* – org. nemá funkčné uzly ale využíva prostriedky ostatných

6.3 Využitie monitoringu

Pri dohliadaní na sieťové prvky sú v sieti PlanetLab k dispozícii dve hlavné monitorovacie rozhrania prístupné cez HTML rozhranie služby CoMon. Sú tvorené naformátovanými a graficky prezentovanými dátami získanými z démonov spustených na každom uzle v sieti. V tejto časti uvediem zopár zaujímavých štatistík sliceu `princeton_comon`, pod ktorým beží samotný monitorovací systém. Ďalej sa budem venovať organizácii CESNET. Jedná sa o prevádzkovateľa výskumnej a vzdelávacej siete, pod ktorú spadá aj sieť VUT v Brne. Ako posledné uvediem zopár problémových uzlov zachytených pri používaní monitorovacej služby CoMon a spôsob jednoduchého vytvorenia vlastnej štatistiky zo získaných dát.

6.3.1 Slice `princeton_comon`

Príloha 11 obsahuje vybrané hodnoty, grafy a zobrazenia týkajúce sa uvedeného sliceu.

Horná časť *Obr. 11.1* znázorňuje priemer hodnôt využívaných prostriedkov zaokrúhlený na dve desatinne miesta. Uvedené hodnoty sú stanovené z intervalu posledných dvoch dní a odzrkadľujú kvalitu monitorovacej služby. Hodnoty prenosu dát, počet spustených procesov a zaťaženie procesoru vypovedajú o vyladení tejto služby. Hodnota potrebnej pamäte je mierne nad hranicou jej priemerného využitia v sieti PlanetLab a preto je zvýraznená oranžovou farbou. Toto zvýšenie je ale opodstatnené testovaním využitia pamäte monitorovacím procesom, ktorý požaduje pridelenie 100MB pamäte na každom uzle. Jeho pridelená priorita je ale nízka a preto v prípade, že na uzle bežia dôležitejšie procesy, je hodnota pridelennej pamäte nižšia. To poskytuje užívateľovi prehľad

o prístupných pamäťových prostriedkoch uzlu pomocou hodnoty „Free Mem“ uvedenej v režime zobrazenia podľa uzlov.

Pri zobrazení problémových sliceov som ale natrafil aj na problém zdanlivo súvisiaci s týmto sliceom a ako príklad som vybral uzol uvedený v druhej tabuľke spolu s hodnotami využitia jeho prostriedkov. Hodnota počtu procesov a využitej pamäte je pomerne vysoká. Zobrazené grafické priebehy lineárnym nárastom jednotlivých veličín jasne ukazujú na nekorektné ukončovanie spustených úloh. Následné zobrazenie výpisu aktívnych sliceov pomocou služby CoTop naznačuje nesprávnu funkčnosť samotného systému a rozšírenie problému aj do ostatných aktívnych sliceov na tomto uzle.

Obr. 11.2 znázorňuje vizualizácie efektivity využitia pamäte v závislosti na množstve prenesených dát. Ako je vidno z farebného označenia odpovedajúcej časti, uvedený pomer je vzhľadom na priemer v sieti PlanetLab stále považovaný za jeden z najlepších. Nasledujúca vizualizácia v pravej hornej časti obrázku znázorňuje priemerné využitie pamäte a počet jeho smerodajných odchýlok od týždňového priemeru. Aktuálne využitie pamäte bolo teda v sledovaný okamih o polovicu smerodajnej odchýlky nad priemer. V dolnej časti obrázku je zobrazená časť vizualizácie počtu uzlov poskytovaných univerzitou v Princetone spolu s príslušnou tabuľkou. Žltá farba pre zodpovedajúcu časť vo vizualizácii znázorňuje aktuálny stav funkčných uzlov. Všetky dostupné uzly mali podľa zobrazenia parametrov uzlov problém s preťažením oboch DNS serverov. Tabuľka ďalej zobrazuje štatistiku uzlov, sliceov, sliverov a aktivitu a označenie organizácie.

Tab. 11.1 je tabuľka spoločná pre vizualizácie efektivity a využitia prostriedkov siete. Obsahuje hodnoty aktuálneho celkového, maximálneho a priemerného využitia prenosového pásma, počtu procesov, využitia fyzickej/virtuálnej pamäte a procesora spolu s hodnotami a počtom ich smerodajných odchýlok od týždňového priemeru. Na konci tabuľky sú tiež uvedené hodnoty týždňového priemeru, smerodajných odchýlok a ich počtu od predchádzajúcej týždňovej štatistiky pre využitie prenosového pásma, pamäte a procesoru.

6.3.2 Výskumná sieť CESNET

Obr. 11.3 v hornej časti znázorňuje problém, na ktorý som natrafil pri testovaní jedného zo zvolených obslužných programov. V sledovanom období boli dva uzly z celkovo dostupných troch nefunkčné. Užívateľské podmienky siete PlanetLab udávajú minimálny počet dvoch funkčných uzlov na organizáciu. Pri dlhodobom porušovaní tejto podmienky dôjde k zakázaniu prístupu pre celú organizáciu až do napravenia vzniknutého problému.

Nasledujúca vizualizácia zobrazuje počet dostupných uzlov a ich stav. Zvolená farba opäť odpovedá aktuálnemu stavu uzlov.

V dolnej časti obrázku je zobrazená štatistika maximálnych nameraných hodnôt pre slice cesnet_s2, ktorý využívam. Všetky hodnoty okrem zaťaženia procesora sú pre kľudový stav v norme. Pri zobrazení grafu histórie pre procesor je vidno, že uvedený stav je dlhodobý. Následne som si zobrazil výpis uzlov patriacich do sliceu a pomocou služby CoTop som si preveril aktuálne využitie prostriedkov sliceom na najviac zaťažovanom

uzle pl2.csl.utoronto.ca. Takéto zaťaženie môže byť napr. spôsobené stále spustenou službou zameranou na určitý druh výpočtu ale taktiež môže ísť o neodladenú službu vo fáze vývoja.

Na nasledujúcom *Obr. 11.4* je výber zo základného zobrazenia výpisu uzlov pre uzly organizácie CESNET vrátane uzlov spadajúcich pod projekt PlanetLab Europe. Dva uzly odmietli spojenie čo naznačuje filtrovanie portov potrebných pre služby monitoringu. Na prvom uvedenom uzle planetlab1.cesnet.cz je spustený väčší počet procesov náročných na výpočtový výkon. Túto skutočnosť znázorňuje hodnota „Busy CPU“ a „1/5 Min Load“. Hodnoty priemerného počtu spustených procesov sú aktuálne nad hranicou dvojdnového priemeru pre daný uzol. Taktiež je vidieť nároky procesov na prenosové pásmo siete. Uzly s predponou „ple“ majú menšie problémy s presnosťou plánovania a na druhom z uzlov bol za posledných 5 minút zaznamenaný výpadok spojenia. Problémy s plánovaním znázorňujú zvýšené hodnoty intervalov medzi dvoma testovacími časovačmi na oboch uzloch. Výpadok spojenia na druhom uzle pravdepodobne naznačuje problém s démonom slúžiacim na kontrolu času potrebného na vytvorenie spojenia alebo krátkodobý problém so sieťovým pripojením pretože hodnoty odozvy na požiadavky monitorovacej služby nie sú zvýšené.

6.3.3 Rôzne problémy v sieti

V hornej časti *Obr. 11.5* je zobrazená základná štatistika uzlu, ktorý je práve využívaný distribučnou sieťou CoDeeN. Táto skutočnosť je zobrazená číselne, graficky a aj výpisom sliceov aktuálne využívajúcich prostriedky uzlu. Distribúciu súboru znázorňuje prenos dát dvoma sliceami distribučnej siete „princeton_coblitz/codeen“. Tiež je vidieť nízku náročnosť distribučného systému na prostriedky uzlu.

Zvolený uzol je ale zároveň postihnutý dvoma problémami. Systémový čas uzlu nie je správne synchronizovaný, vid' hodnota „Drift“ a slice „gt_pithia“ zobrazený vo výpise nadmerne zaťažuje procesor uzlu zvýšeným počtom spustených procesov. Podozrivé je hlavne nekorektné ukončovanie týchto procesov čo je viditeľné na veľkom počte spiacich procesov tohto sliceu. Na uzle je taktiež veľký nápor na rozdeľovanie pamäte pretože testovací démon obdržal z požadovaných 100 MB iba 59 znázornených premennou „Free Mem“.

V dolnej časti obrázku je zobrazený uzol s nefunkčným SSH prihlasovaním a vysokým reakčným časom na požiadavky monitorovacej služby. Vysoký počet spustených procesov a z toho plynúca systémová záťaž procesora ukazujú na nesprávnu funkčnosť systému uzlu. Túto skutočnosť potvrdzuje aj chyba monitorovania virtuálnych serverov resp. nefunkčnosť služby CoTop na tomto uzle, ktorá práve z uvedeného dôvodu nemohla byť využitá na bližšiu identifikáciu vzniknutého problému. Uvedený uzol má taktiež vysokú úroveň chybovosti DNS požiadaviek.

Obr. 11.6 obsahuje výber z rozšíreného zobrazenia služby CoMon zameraný na parametre zobrazujúce názvy sliceov, ktoré sú najväčšími užívateľmi prostriedkov daných uzlov. Na prvom uzle je pod sliceom „google_highground“ spustený nadmerný počet

procesov, ktoré spotrebúvajú väčšinu dostupnej pamäte. Na ďalšom uzle je napr. vidieť nadmernú záťaž procesoru spôsobenú root účtom. Môže to naznačovať napr. prebiehajúcu aktualizáciu alebo aj nekorektne nakonfigurovanú službu spustenú na pozadí. Posledný stĺpec znázorňuje názvy sliceov, ktoré na jednotlivých uzloch najviac využívajú prenosové pásmo čo urýchľuje odhalenie nadmerného zaťaženia siete a jeho príčiny.

6.3.4 Vlastný výber a spracovanie dát

Pomocou vzorových požiadaviek pre HTML rozhranie služby CoMon je možné rýchlo a jednoducho získať potrebné dáta podľa vlastného výberu. Formátovanie a popis jednotlivých požiadaviek je uvedený v kapitolách 6.1.2 a 6.1.3.

Príkladom je *Tab. 11.2*. Na jej zhotovenie boli použité zvolené požiadavky vyžiadané vo formáte CSV príkazom „format = formatcsv“ a následným výberom konkrétnych hodnôt príkazom „dumpcols“. Alternatívnym príkazom na delenie je príkaz „format = formatspaces“, ktorý oddeľuje hodnoty tabelovanými medzerami. Použitý formát CSV prevedie údaje do neformátovaného textu s hodnotami delenými do stĺpcov pomocou čiarky. Tento text sa dá, po prekopírovaní z okna prehliadača do textového súboru, importovať napr. do programu Excel pre vytvorenie požadovaných grafov, prípadne na následné formátovanie údajov. Pri importe sa program spýta na spôsob delenia hodnôt kvôli ich správne mu zoradeniu do stĺpcov, na počiatočný riadok tabuľky, typy údajov v jednotlivých stĺpcoch a na ďalšie upresňujúce voľby. Na získanie hodnôt do uvedenej tabuľky som zostavil nasledujúcu požiadavku pre službu CoMon:

```
http://comon.cs.princeton.edu/status/tabulator.cgi?table=table_nodeviewsh  
ort&sort=resptime&select='(resptime < 0.1&&resptime > 0&&lminload <  
5&&gbfree > 80&&numcores >= 4&&(dnsludp < 20 || dns2udp <  
20))'&format=formatcsv&dumpcols='name,resptime,sshstatus,cpuspeed,numcore  
s,freecpu,lminload,numslices,memsize,memact,freemem,disksize,gbfree,bwlim  
it,txrate,rxrate'
```

Dáta importované do programu Excel boli následne naformátované a upravené podľa potreby.

7 ZÁVER

Pri práci na zvolenej téme bakalárskej práce som sa zoznámil so štruktúrou a jednotlivými prvkami výskumnej siete PlanetLab. Na základe získaných informácií a dostupných materiálov som spracoval postup vytvorenia prístupu do siete a zásady používania pridelených prostriedkov.

Po vytvorení vlastného prístupu som vybral z dostupného programového vybavenia vhodné aplikácie spĺňajúce požiadavky uvedené v cieľoch práce. Následne som pre zvolené aplikácie vypracoval návod na inštaláciu a konfiguráciu pre linuxovú distribúciu Ubuntu 10.10 x64, popis ich prostredia a jednotlivých funkcií ako aj možnosti ich praktického využitia v prostredí siete.

Na prácu v príkazovom riadku som pre jeho jednoduchosť a univerzálnosť použil program CoDeploy. Uvedený postup inštalácie bol venovaný hlavne správne nastaveniu prostredia terminálu pre bezproblémový chod programu. V závere kapitoly som uviedol využitie programu na paralelné spúšťanie príkazov a upload súborov na viacero uzlov súčasne pre potreby aplikácie na lokalizovanie polohy organizácií v sieti PlanetLab.

Dve nasledujúce časti obsahujú vypracovaný manuál k programu Plush a jeho grafickej nadstavbe Nebula. Uvedené sú potrebné knižnice a ich verzie použité pri inštalácii, možnosť aktualizácie pomocných skriptov programu v staršom skompilovanom balíčku a príprava a postup samostatnej kompilácie. Pri oživovaní programu som komunikoval s jeho vývojárom a riešil problémy s funkčnosťou programu v distribúcií Ubuntu. Toto úsilie viedlo k aktualizácii skompilovaných balíčkov programu a umožnení jeho využitia aj v novších distribúciách Linuxu bez nutnosti samostatnej kompilácie. Po úspešnej kompilácii bolo prekonzultované a následne aj spracované správne nastavenie programu. Kapitola grafickej nadstavby obsahuje riešenie problému s prostredím opengl v Jave a s SSH klientom Mindterm, ktorého Nebula využíva. Obsahom je aj nastavenie grafického prostredia. Programy Plush a Nebula som zvolil z dôvodu možnosti vývoja vlastných XML aplikácií a ich upload a testovanie v sieti bez nutnosti využitia prostredia príkazového riadku. V uvedenom prostredí som vytvoril aplikáciu na lokalizáciu organizácií z časti venovanej programu CoDeploy.

Nasledujúca časť bola venovaná softvérovej alternatíve PIMan. Vypracovaný bol návod pre Ubuntu ako aj pre systém Windows 7 Professional x64. V systéme Windows 7 bolo vypracované riešenie nekorektnej implemetácie aktuálnej verzie Javy. Taktiež bol uvedený zdroj a postup inštalácie pre chýbajúci balíček grafických nástrojov jazyku Java v distribúcií Ubuntu.

V závere práce bol spracovaný popis prostredia monitorovacieho systému CoMon a CoVisualize. Tento systém bol následne využitý na monitorovanie činnosti uzlov a odhalenie prípadných problémov. Uvedené boli aj možné spôsoby získania dát a ich následné spracovanie.

Na priloženom médiu sa nachádzajú verzie knižníc použité pri inštalácii jednotlivých programov, všetky softvérové balíčky, s ktorými som pracoval, dodatočný softvér, ktorý

bol potrebný pri riešení problémov, podklady a samotná aplikácia z praktickej časti a ostatné použité materiály.

POUŽITÁ LITERATÚRA

- [1] ALBRECHT Jeannie a kolektív UCSD. *Plush* [online]. [citované 2011-4-1]. Dostupné z: < <http://plush.cs.williams.edu/index.html> >.
- [2] CHUPÁČ, M. *Obsluha stanic Planetlabu*. Semestrálna práca. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011.
- [3] FIUCZYNSKI E. M. *User's Guide* [online]. 02/01/2006 [citované 2011-5-14]. Dostupné z: < <https://www.planet-lab.org/doc/guides/user/> >.
- [4] ISDAL Tomas a kolektív. *PlMan* [online]. 16/08/2007 [citované 2011-5-23]. Dostupné z: < <http://www.cs.washington.edu/research/networking/cplane/> >.
- [5] PAI Vivek, PARK KyoungSoo a kolektív. *CoMon* [online]. [citované 2011-4-3]. Dostupné z: < <http://comon.cs.princeton.edu> >.
- [6] PAI Vivek, PARK KyoungSoo, PETERSON Larry a kolektív. *CoDeploy* [online]. [citované 2011-3-21]. Dostupné z: < <http://codeen.cs.princeton.edu/codeploy/> >.
- [7] SAENG SUWARN Sratha, PAI Vivek. *CoVisualize* [online]. [citované 2011-4-4]. Dostupné z: < <http://codeen.cs.princeton.edu/covisualize/> >.
- [8] TMACK. *Hello World Tutorial* [online]. 20/10/2009 [citované 2011-3-21]. Dostupné z: < <http://svn.planet-lab.org/wiki/HelloWorldTutorial/> >.
- [9] UCSD kolektív. *Plush/Nebula Tutorial* [online]. [citované 2011-5-14]. Dostupné z: < <http://plush.cs.williams.edu/nebula/tutorial.html> >.

ZOZNAM SKRATIEK A JEDNOTIEK

CGI	– Computer-generated Imagery – Počítačová animácia resp. počítačom generovaná grafika a efekty.
CSV	– Comma-separated Values – Hodnoty oddelené čiarkami, súborový formát určený na výmenu tabuľkových dát.
DNS	– Domain Name System – Hierarchický systém doménových mien, distribuovaná databáza sieťových informácií prenášaná pomocou rovnomenného protokolu.
FAQ	– Frequently Asked Questions – Často kladené otázky.
GB	– Gigabajt – Označenie jednotky množstva informácie, 10^9 bajtu.
GHz	– Gigahertz – Jednotka frekvencie, 10^9 Hertz.
HTML	– HyperText Markup Language – Znakový jazyk pre hypertext.
HTTP/S	– Hypertext Transport Protocol / Secure – Hypertextový transportný protokol určený na prenos dokumentov vo formáte HTML / jeho zabezpečená nadstavba.
IDS	– Intrusion Detection System – Systém detekcie škodlivého chovania / narušiteľa.
ICMP	– Internet Control Message Protocol – Protokol IP sietí, využíva sa hlavne na zasielanie správ pri chybách v IP datagramoch alebo pre diagnostické a routovacie účely.
kb	– kilobit – Tisíc násobok základnej jednotky informácie bit.
kbps	– kilobit per second – Jednotka prenosovej rýchlosti udávajúca počet tisícov bitov prenesených za sekundu.
LMB/RMB	– Left/Right Mouse Button – Ľavé/Pravé tlačidlo myši.
MB	– Megabajt – Označenie jednotky množstva informácie, 10^6 bajtu.

ms	– milisekunda – Tisícina základnej jednotky času sekunda.
PHP	– Hypertext PreProcessor – Hypertextový preprocesor, skriptovací programovací jazyk.
RSA	– Šifra s verejným kľúčom (iniciály autorov Rivest, Shamir, Adleman)
SSH	– Secure Shell – Zabezpečený komunikačný protokol v počítačových sieťach a zároveň názov terminálu, ktorý tento typ zabezpečenia používa.
tar	– tape archiver – Program na kompresiu súborov, súborový formát, prípona výstupného súboru programu má rovnaké označenie.
URL	– Uniform Resource Locator – Jednotný lokátor zdroja, reťazec znakov z definovanou štruktúrou, ktorý slúži na špecifikáciu umiestnenia zdroja na internete.
XML	– eXtensible Markup Language – Rozšíriteľný značkovací jazyk.
XML-RPC	– eXtensible Markup Language – Remote Procedure Call – Protokol vzdialeného volania procedúry, dáta prenášané týmto protokolom sú vo formáte XML.

ZOZNAM PRÍKAZOV

Príkaz 2.1: Vytvorenie SSH kľúča.	12
Príkaz 2.2: Prihlásenie k vzdialenému uzlu.	13
Príkaz 2.3: Riešenie problémov s prihlásením.	14
Príkaz 2.4: Upload súborov pomocou nástroja scp.	15
Príkaz 2.5: Aktualizácia súborov nástrojom rsync.	16
Príkaz 2.6: Povolenie skriptu pomocou nástroja chkconfig.	16
Príkaz 2.7: Spustenie/ukončenie skriptu nástrojom service.	16
Príkaz 3.1: Zrušenie overovania kľúču pri prihlasovaní.	23
Príkaz 3.2: Nastavenie premenných prostredia pre program CoDeploy.	23
Príkaz 3.3: Syntax príkazu programu CoDeploy.	25
Príkaz 3.4: Syntax príkazu programu Multiquery.	25
Príkaz 3.5: Syntax využívaná programom Multicopy.	25
Príkaz 3.6: Príkaz na inštaláciu knižnice.	32
Príkaz 3.7: Download a inštalácia knižnice xmlrpc.	32
Príkaz 3.8: Download a kompilácia programu Plush.	34
Príkaz 3.9: Spustenie konfiguračného skriptu programu Plush.	35
Príkaz 3.10: Nastavenie programu Plush.	35
Príkaz 3.11: Spustenie testovacieho skriptu programu Plush.	36
Príkaz 3.12: Inštalácia modulov Frontier::Client a Crypt::SSLeay nástrojom cpan.	36
Príkaz 3.13: Overenie funkčnosti programu Plush.	37

ZOZNAM OBRÁZKOV

Obr. 2.1: Celosvetové rozšírenie organizácií PlanetLabu.	10
Obr. 3.1: Terminálové prostredie programu.	20
Obr. 3.2: Terminálové prostredie Plushu.	29
Obr. 3.3: Popis funkcie Plush kontroléru.	30
Obr. 4.1: Grafické prostredie Nebuly.	40
Obr. 5.1: Grafické prostredie programu PIMan.	52
Obr. 6.1: HTML prostredie služby CoMon.	58
Obr. 6.2: Prostredie vizualizácií.	66

ZOZNAM PRÍLOH

Skript 8.1: Python skript na pridanie uzlov k sliceu [8].	80
Skript 8.2: Obsah súboru „phonelog.php“ [8].	80
Skript 8.3: Obsah súboru „phonehome.php“ [8].	81
Skript 8.4: Obsah súboru „phonehome.py“ [8].	81
Obr. 8.1: Zobrazenie súboru „KML“.	83
Obr. 9.1: Základné zobrazenie mapy sveta.	84
Obr. 9.2: Monitorovanie uzlov.	84
Obr. 9.3: Grafické zobrazenie sledovaného parametru.	85
Obr. 9.4: Zoznam uzlov na pridanie.	85
Obr. 9.5: Prihlásenie k uzlu.	86
Obr. 9.6: Nastavenie komponentu.	86
Obr. 9.7: Nastavenie procesu.	86
Obr. 9.8: Zobrazenie „HelloWorld“ aplikácie.	87
Obr. 9.9: Nastavenie SWORD vyhľadávania.	87
Obr. 9.10: Prostredie zadávania príkazov [9].	88
Obr. 9.11: Vytvorený komponent.	88
Obr. 9.12: Využitie synchronizačných prvkov.	89
Obr. 9.13: Výstup programu Plush.	89
Obr. 9.14: Vlastná aplikácia a zoznam priradených prostriedkov.	90
Obr. 9.15: Zobrazenie stavu vykonávania aplikácie a procesy klienta na vzd. uzle.	90
Obr. 10.1: Rozhranie výberu uzlov na pripojenie.	91
Obr. 10.2: Paralelné spúšťanie príkazov a terminálový výstup.	92
Obr. 10.3: Funkcia sťahovania a jej priebeh.	93
Obr. 11.1: Využitie monitorovacej služby CoMon a Cotop.	94
Obr. 11.2: Využitie modulu vizualizácií.	95
Obr. 11.3: Využitie vizualizácie a monitorovacích služieb CoMon a CoTop.	97
Obr. 11.4: Uzly organizácie CESNET.	98
Obr. 11.5: Využitie služby CoMon na odhalenie možných problémov.	99
Obr. 11.6: Parametre rozšíreného zobrazenia služby CoMon.	100
Tab. 11.1: Tabuľka hodnôt potrebných na zhotovenie vizualizácií.	96
Tab. 11.2: Tabuľka vytvorená zo zvolených dát vo formáte CSV.	101

8 PRVÁ PRÍLOHA

Skript 8.1: Python skript na pridanie uzlov k sliceu [8].

```
$ python
>>> import xmlrpclib
>>> api_server = \
... xmlrpclib.ServerProxy('https://www.planet-lab.org/PLCAPI/')
>>>
>>> auth = {}
>>> auth['Username'] = "user" <-- zadajte Vaše užívateľské meno
>>> auth['AuthString'] = "pass" <-- zadajte názov Vášho sliceu
>>> auth['AuthMethod'] = "password"
>>>
>>> node_list = [line.strip() for line in open("cesta_k_nodes.txt")]
>>>
>>> api_server.AddSliceToNodes(auth, \
... "your_slice_name", node_list)
>>> ^D
```

Skript 8.2: Obsah súboru „phonelog.php“ [8].

```
<?
### phonelog.php
header('Content-type: application/vnd.google-earth.kml+xml');
header('Content-disposition: attachment;
      filename="hello_world.kml"');
echo '<?xml version="1.0" encoding="UTF-8"?>' . "\n";
echo '<kml xmlns="http://earth.google.com/kml/2.0">' . "\n";
echo '<Document>' . "\n";

$phonelog = file("phonelog.txt");
foreach ($phonelog as $entry) {
    $fields = explode("\t", trim($entry));
    if ($fields[4] == "None" || $fields[5] == "None") continue;
?>
<Placemark>
  <Snippet><?= htmlspecialchars($fields[3]) ?></Snippet>
  <name><?= htmlspecialchars($fields[1]) ?></name>
  <LookAt>
    <longitude><?= $fields[5] ?></longitude>
    <latitude><?= $fields[4] ?></latitude>
    <range>1000000</range>
  </LookAt>
  <visibility>1</visibility>
  <Point>
    <extrude>1</extrude>
    <altitudeMode>relativeToGround</altitudeMode>
```

```

        <coordinates>
            <?=$fields[5] . "," . $fields[4] ?>,0
        </coordinates>
    </Point>
</Placemark>
<? }
?>
</Document>
</kml>

```

Skript 8.3: Obsah súboru „phonehome.php“ [8].

```

<?
### phonehome.php
if (isset($_GET['reset'])) {
    $phonelog = fopen("phonelog.txt", "w");
    fclose($phonelog);
    echo "Cleared Phone Log.";
    die();
}
if (!isset($_POST['site_id'])) die;
$existinglog = file("phonelog.txt");
foreach ($existinglog as $entry) {
    $f = explode("\t", trim($entry));
    if ($f[0] == $_POST['site_id']) die;
}

$phonelog = fopen("phonelog.txt", "a");
fwrite($phonelog, $_POST['site_id'] . "\t" . $_POST['name'] .
    "\t" . $_POST['login_base'] . "\t" . $_POST['url'] .
    "\t" . $_POST['latitude'] . "\t" . $_POST['longitude'] .
    "\n");
fclose($phonelog);
?>

```

Skript 8.4: Obsah súboru „phonehome.py“ [8].

```

#!/usr/bin/python
### phonehome.py
### Hello World demonstration script

phonehome_url = "http://adresa_webserveru/phonehome.php"

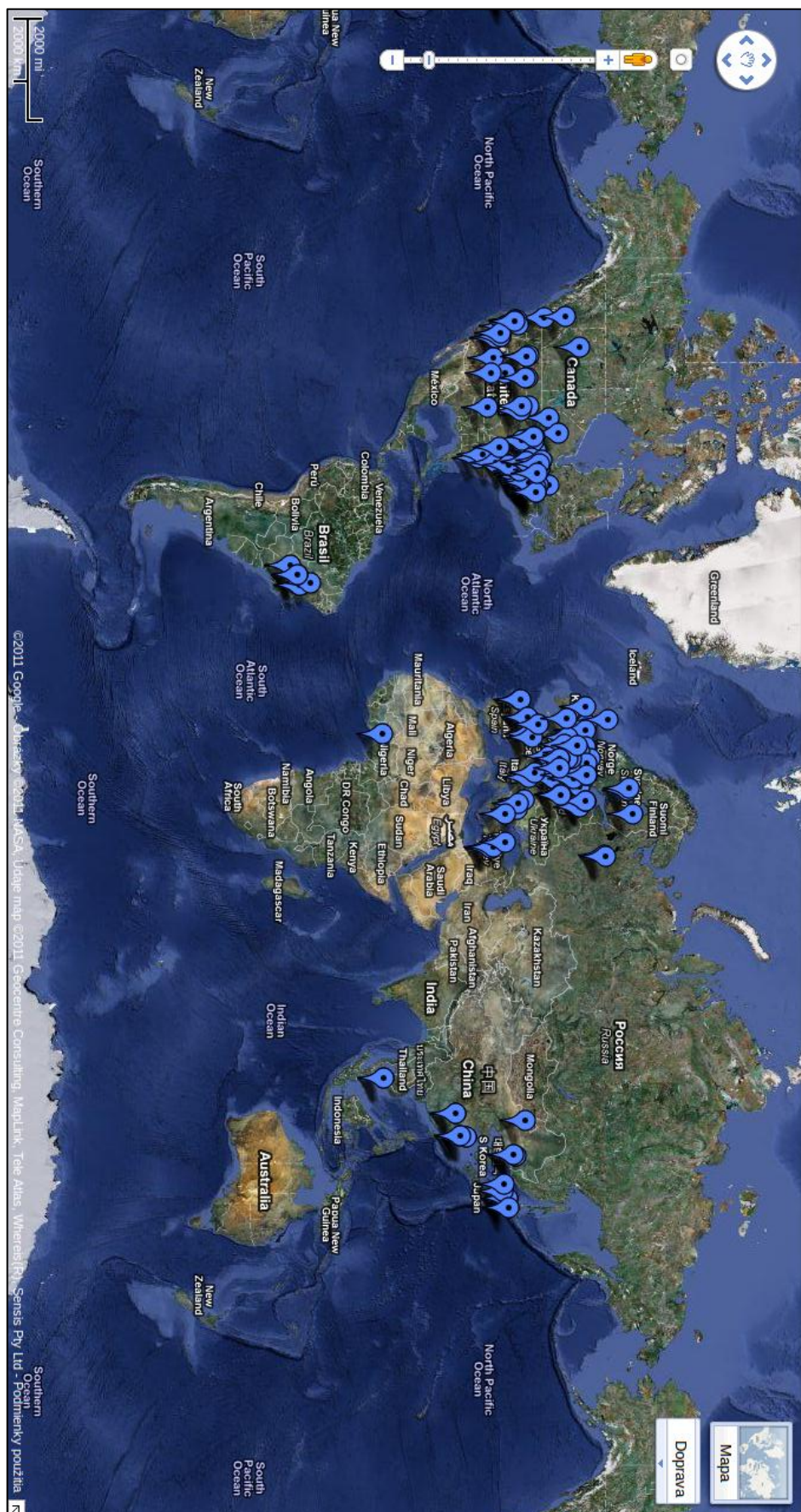
import sys, urllib, xmlrpclib, socket

api_server = xmlrpclib.ServerProxy('https://www.planet-lab.org/PLCAPI/')

```

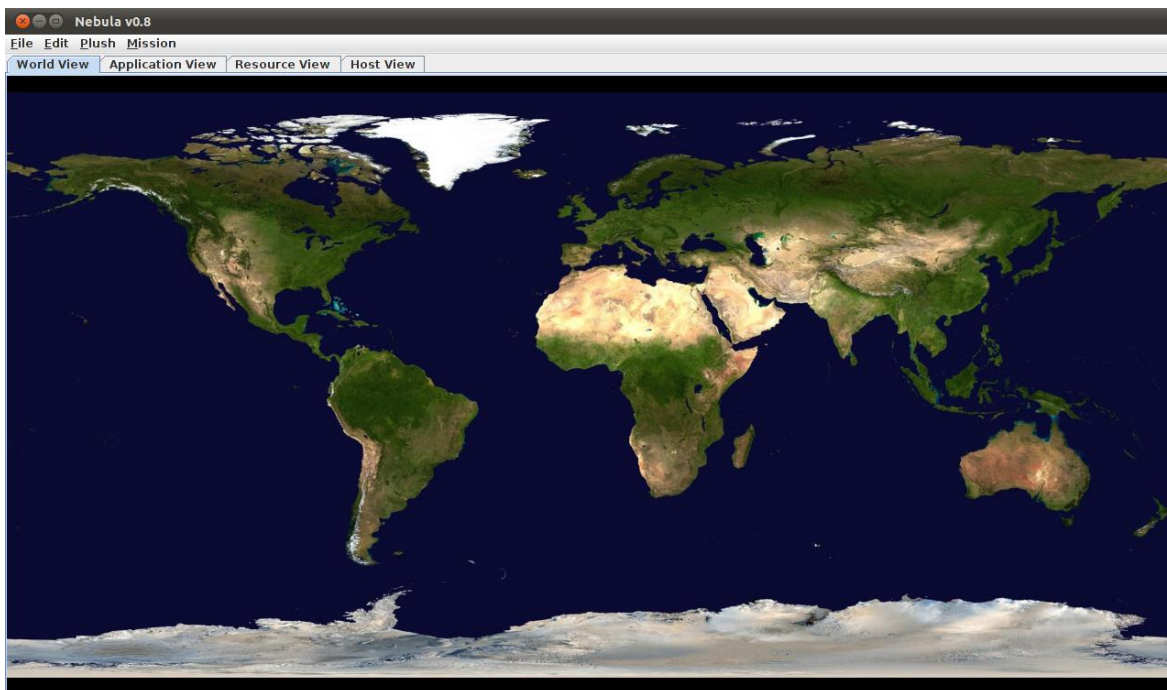
```
auth = {}
auth['AuthMethod'] = "anonymous"
auth['Role'] = "user"

hostname = socket.gethostname()
query = api_server.GetNodes(auth,
                             {'hostname': hostname}, ['site_id'])
site_id = query[0]['site_id']
site_info = api_server.GetSites(auth,
                                 {'site_id': site_id}, ['site_id', 'name', 'url',
                                                         'latitude', 'longitude', 'login_base'])
site_info = urllib.urlencode(site_info[0])
urllib.urlopen(phonehome_url, site_info)
```

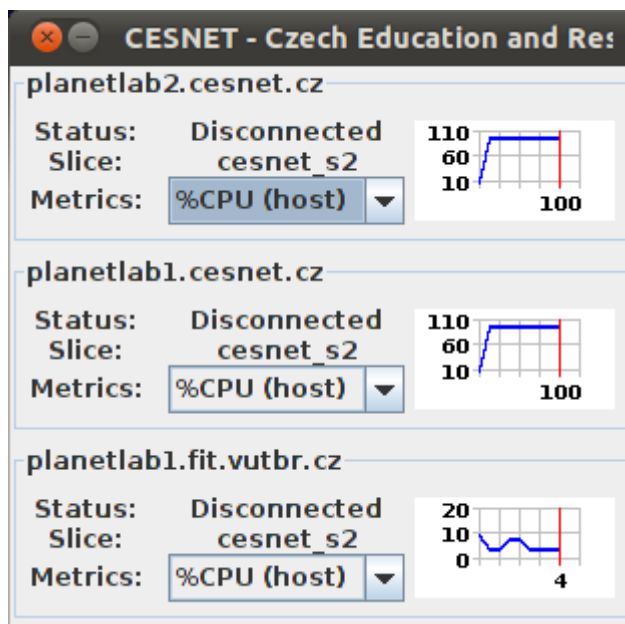


Obr. 8.1: Zobrazenie súboru „KML“.

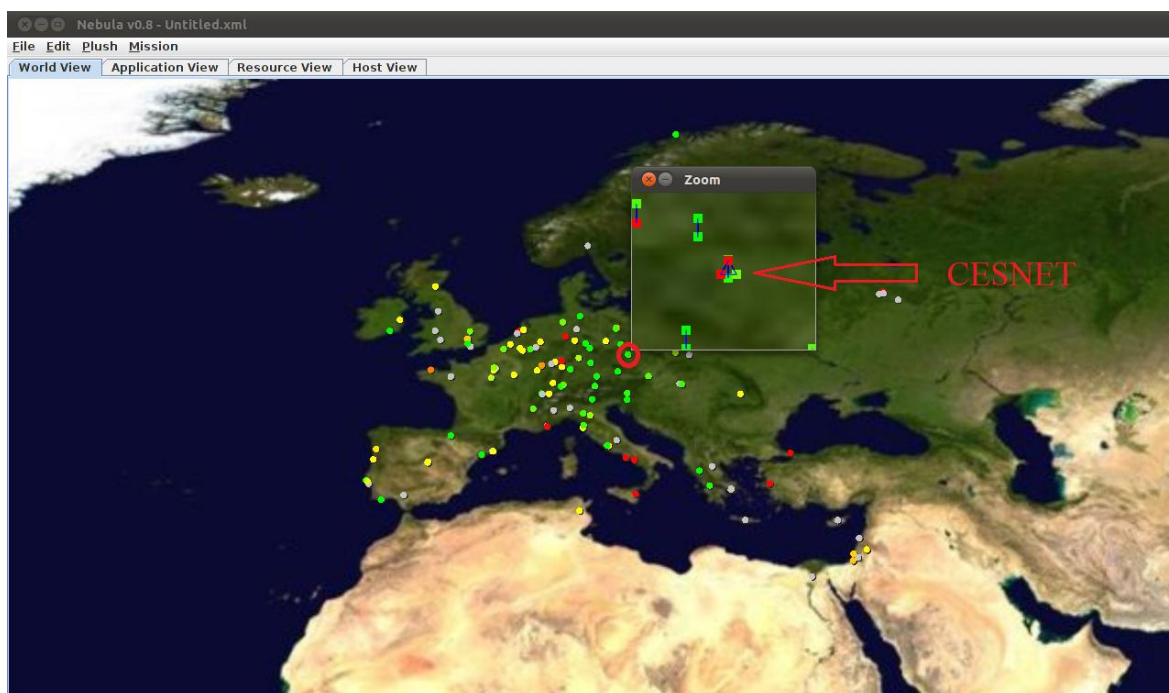
9 DRUHÁ PRÍLOHA



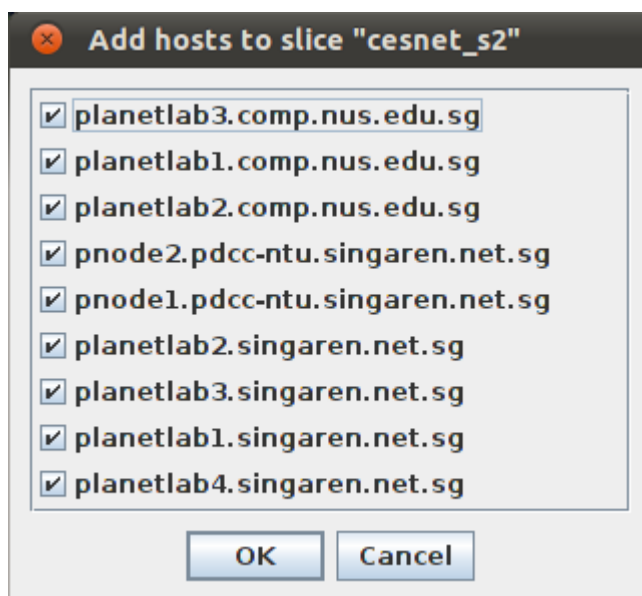
Obr. 9.1: Základné zobrazenie mapy sveta.



Obr. 9.2: Monitorovanie uzlov.



Obr. 9.3: Grafické zobrazenie sledovaného parametru.



Obr. 9.4: Zoznam uzlov na pridanie.

planetlab2.cesnet.cz		Disconnected
planetlab2.itwm.fhg	SSH to planetlab2.cesnet.cz	Disconnected
planet1.scs.cs.nyu.	Fail planetlab2.cesnet.cz	Disconnected

Obr. 9.5: Prihlásenie k uzlu.

Add component block

Component

Name:

Number of hosts:

Resource

Type:

Slice:

Software

Select which software to include:

No software package exists
Please create one

Obr. 9.6: Nastavenie komponentu.

Edit process

Process

Name:

Path:

Command Args:

Predecessor

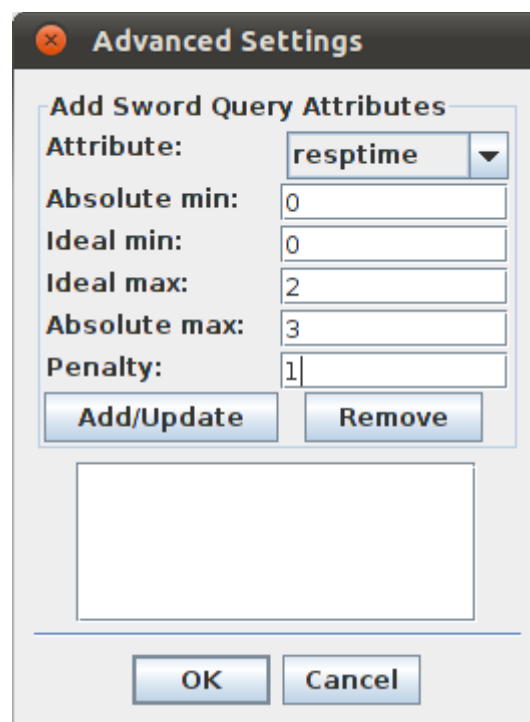
Select and remove predecessor:

None

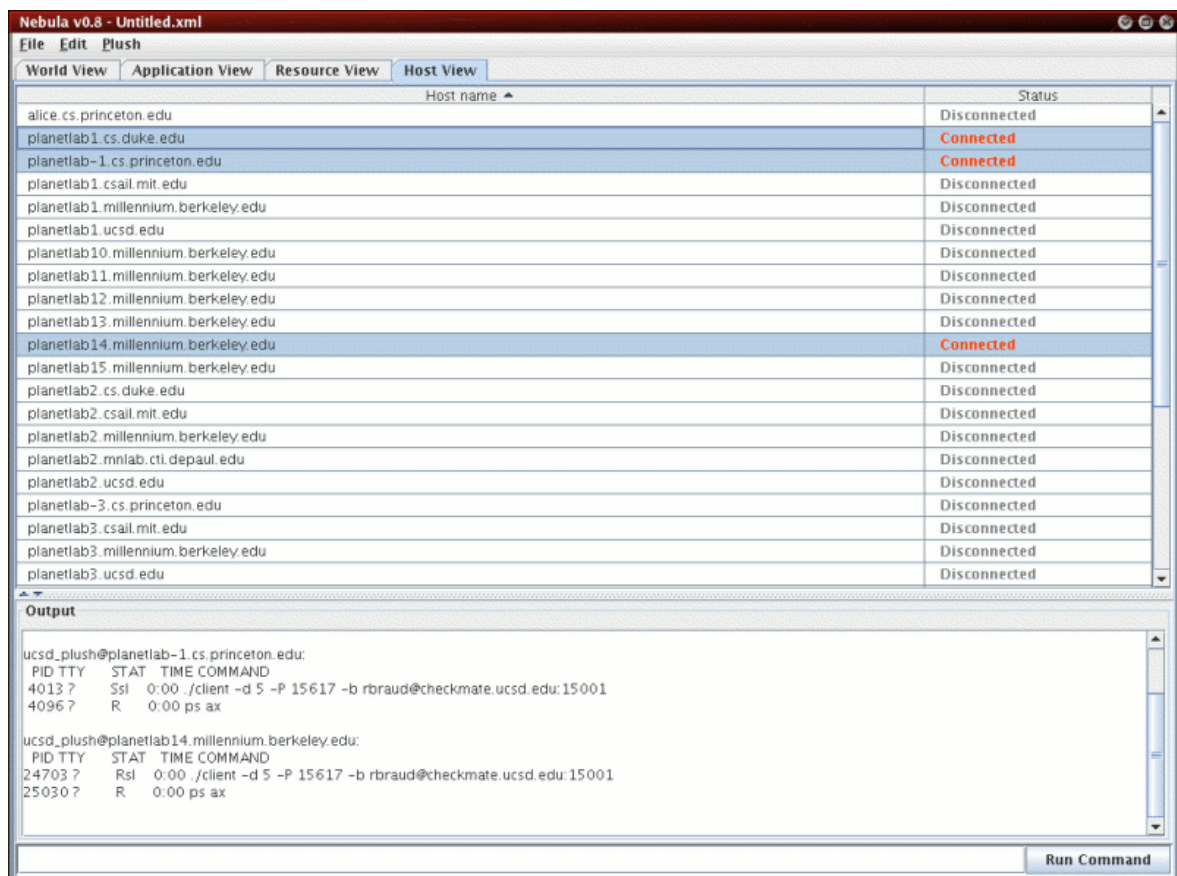
Obr. 9.7: Nastavenie procesu.



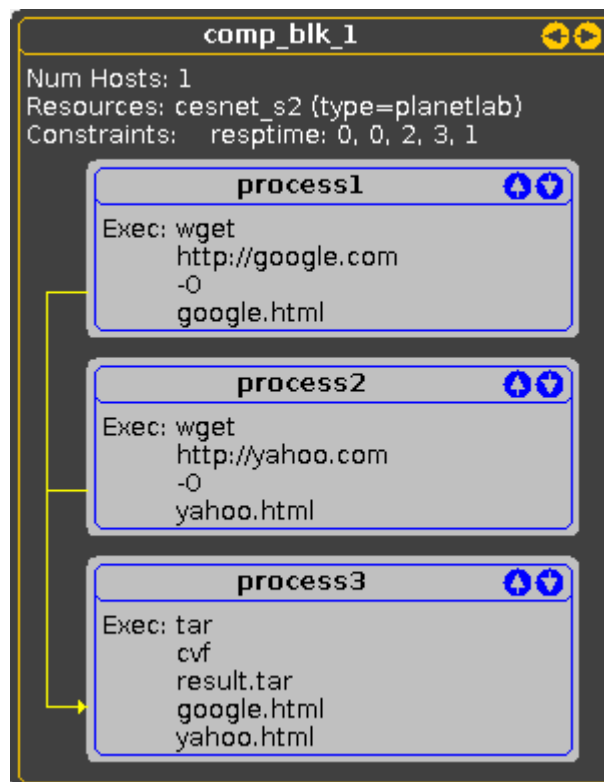
Obr. 9.8: Zobrazenie „HelloWorld“ aplikácie.



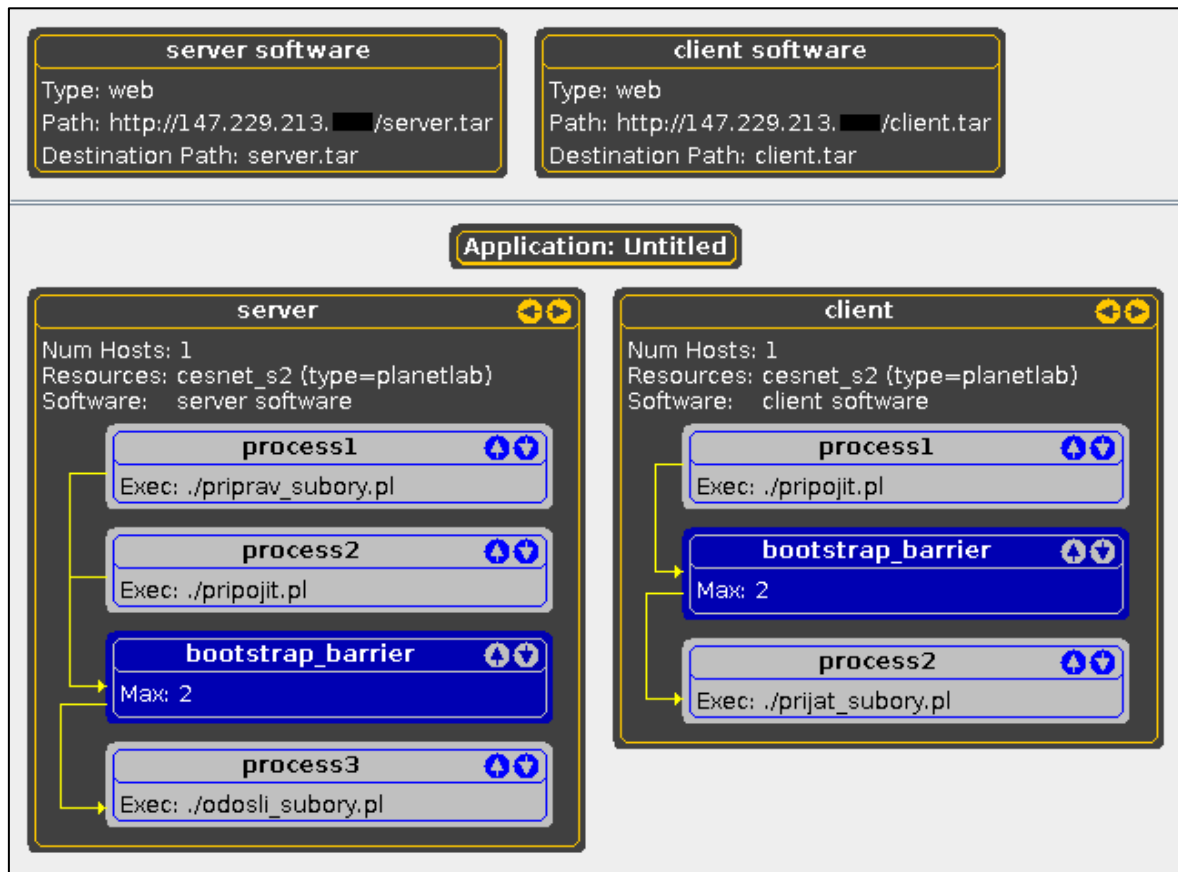
Obr. 9.9: Nastavenie SWORD vyhľadávania.



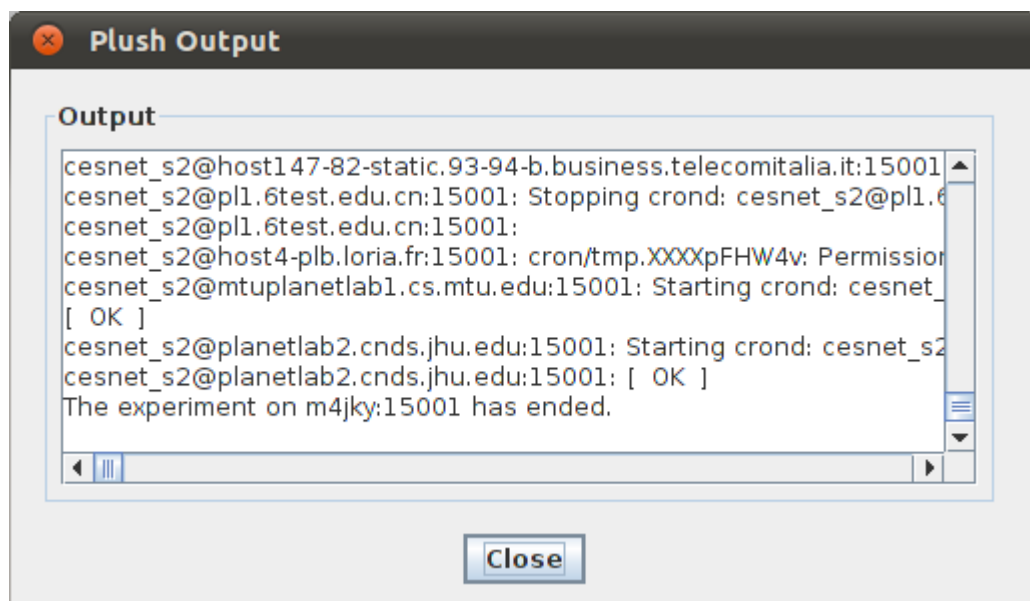
Obr. 9.10: Prostredie zadávania príkazov [9].



Obr. 9.11: Vytvorený komponent.



Obr. 9.12: Využitie synchronizačných prvkov.



Obr. 9.13: Výstup programu Plush.

Application View

Python skript
Type: web
Path: http://147.229.213.█/phonehome.py
Destination Path: phonehome.py

Definicia spustenia skriptu
Type: web
Path: http://147.229.213.█/locate.cron
Destination Path: locate.cron

Application: Urcenie polohy

Aplikacia
Num Hosts: 50
Resources: cesnet_s2 (type=planetlab)
Constraints: resptime:0, 0, 2, 3, 1
oneminload:0, 0, 0, 5, 1
Software: Python skript
Definicia spustenia skriptu

Spustitelnost skriptu
Exec: chmod
+x
phonehome.py

Oznamenie cronu
Exec: crontab
locate.cron

Spustenie cron deamona
Exec: sudo
/sbin/service
crond
start

Cakaj
Exec: sleep
125

Zastavenie cron deamona
Exec: sudo
/sbin/service
crond
stop

Vycistenie crontabu
Exec: crontab
-r

Clean up
Exec: rm
phonehome.py
locate.cron

Resource View

Aplikacia
Slice: cesnet_s2
Num Hosts: 50
Assigned Hosts:
earth.cs.brown.edu
gschembra4.dit.unict.it
host147-92-static.93-94-b.business.tel...
kc-sce-plab2.umkc.edu
kostis.di.uoa.gr
metis.mcs.suffolk.edu
miranda.planetlab.cs.umd.edu
node1.planetlab.mathcs.emory.edu
node2pl.planetlab.telecom-lille1.eu
nodea.howard.edu
pads21.cs.nthu.edu.tw
peeramidion.irisa.fr
pl1.cs.yale.edu
pl1.eng.monash.edu.au
pl1.planetlab.ics.tut.ac.jp
pl1snu.koren21.net
pl2-alblf.homeip.net
pl4.unm.edu
plab1.larc.usp.br
plab1.nec-labs.com
plab2.ple.silweb.pl
plab3.eece.ksu.edu
planet-lab1.cs.ucr.edu
planet1.cs.rochester.edu
planet1.pnl.nitech.ac.jp
planet1.unipr.it
planetlab-01.bu.edu
planetlab-01.vt.nodes.planet-lab.org
planetlab-1.cs.uic.edu
planetlab-1.cs.unibas.ch
planetlab-1.elisa.cpsc.ucalgary.ca
planetlab-1.sjtu.edu.cn
planetlab01.tkn.tu-berlin.de
planetlab2.pop-rj.rnp.br
planetlab2.willab.fi
plgmu4.ite.gmu.edu
plink.cs.uwaterloo.ca
plnode-03.gpolab.bbn.com
plnode02.cs.mu.oz.au
publ1-s.ane.cmc.osaka-u.ac.jp
ricepl-4.cs.rice.edu
ricepl-5.cs.rice.edu
righthand.eecs.harvard.edu
roam1.cs.ou.edu
roam2.cs.ou.edu
server4.planetlab.iit-tech.net
thalescom-48-42.cnt.nerim.net
ttu1-1.nodes.planet-lab.org
venus.silicon-valley.ru
vn5.cse.wustl.edu

Obr. 9.14: Vlastná aplikácia a zoznam priradených prostriedkov.

World View

Host name		Status
pads21.cs.nthu.edu.tw		Connecting
rochefort.infonet.fundp.ac.be		Connected
plnode01.cs.mu.oz.au		Installing software
plnode02.cs.mu.oz.au		Setting up application
plab1.larc.usp.br		Running
planetlab-1.cs.uic.edu		Done
evghu7.colbud.hu		Failed

Procesy Plush klienta na vzdialenom uzle:

```

[cesnet_s2@pl1snu ~]$ ps auxwww
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
980        21497  0.0  0.0   4956  2964 ?        Ss   May03   0:00 perl ./bootstra
p.pl --client prefix
980        21507  0.0  0.0   4984  1520 ?        S    May03   0:00 wget -q -O - ht
tp://plush.cs.williams.edu/dist/main/VERSION

[cesnet_s2@planetlab2 ~]$ ps auxwww
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
554        794  0.0  0.0  19292  2016 ?        Ssl  00:08   0:00 ./client -c ./
-d 5 -P 15001 -b m4jky@m4jky:15001

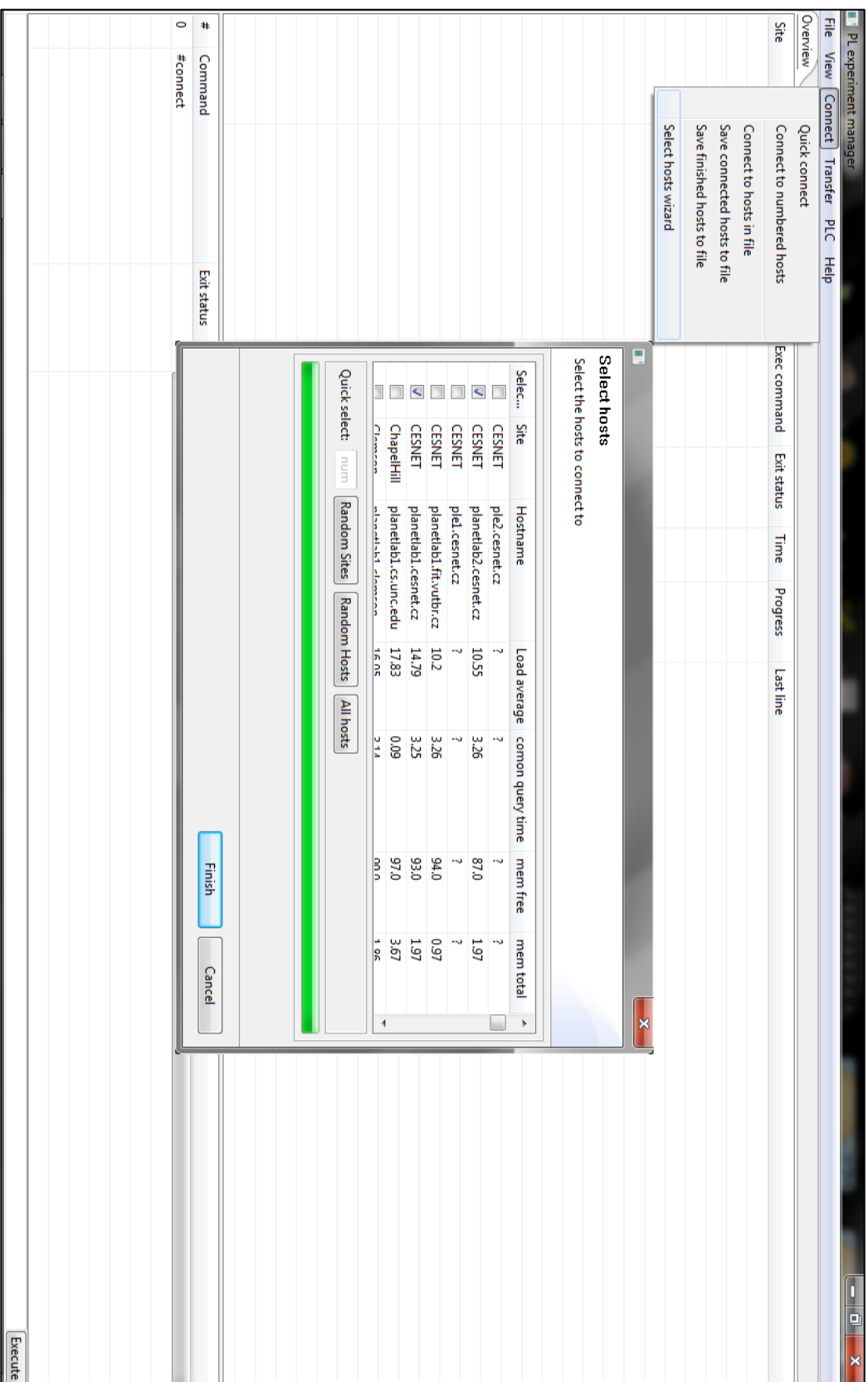
```

World View



Obr. 9.15: Zobrazenie stavu vykonavania aplikácie a procesy klienta na vzd. uzle.

10 TRETIA PRÍLOHA



Obr. 10.1: Rozhranie výberu uzlov na pripojenie.

PL experiment manager

File View Connect Transfer PLC Help

Overview

planetlab1.cesnet.cz

planetlab2.cesnet.cz

Site	Hostname	Exec command	Exit status	Time	Progress	Last line
CESNET	planetlab1.cesnet.cz	ps auxwww	0	9.6		killed after 9.6 s
CESNET	planetlab2.cesnet.cz	ps auxwww	0	0.1		554 32361 0.0 0.0 2632 1220 ? S 22:07 0:00 -bash

File View Connect Transfer PLC Help

Overview planetlab1.cesnet.cz planetlab2.cesnet.cz

Exec command

#connect

ls

ps auxwww

Exit status

0

0

0

Time

0.9

0.1

0.1

```

> #connect
connect successful
> ls
BP_C2_2008.pdf
Unit1.c
bootstrap.pl
client
client.txt
gnp
hello
helper-scripts
install_yum.sh
logfile-planetlab2-15001-1305980706.txt
logfile-planetlab2-15001-1306142890.txt
plush-i386-Linux.tar
plush-logfile.txt
plush-logfile15001-1305980706.txt
plush-logfile15001-1306142890.txt
temp0
temp1
temp10
temp3
temp3
vivaldi
vping
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
554 3553 0.0 0.0 2404 680 ? R 22:09 0:00 ps auxwww
root 32293 0.0 0.0 2656 1072 ? Ss 22:07 0:00 /bin/su - cesnet_s2
554 32361 0.0 0.0 2632 1220 ? S 22:07 0:00 -bash

```

#

Command

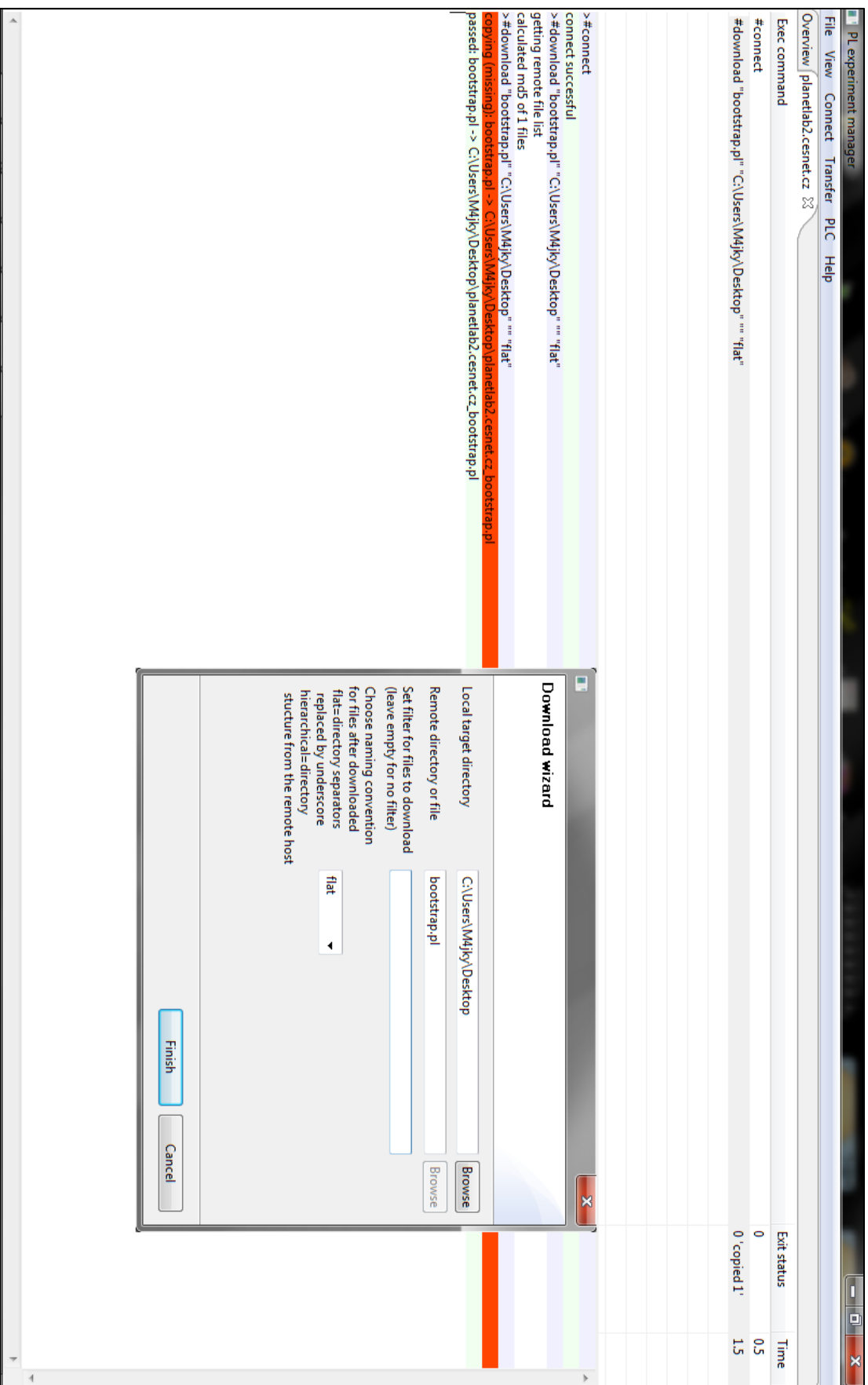
Exit status

Completed hosts

0	#connect	0,2,	
1	ls	-1,1, 0,1,	
2	ps auxwww	-1,1, 0,1,	

Execute

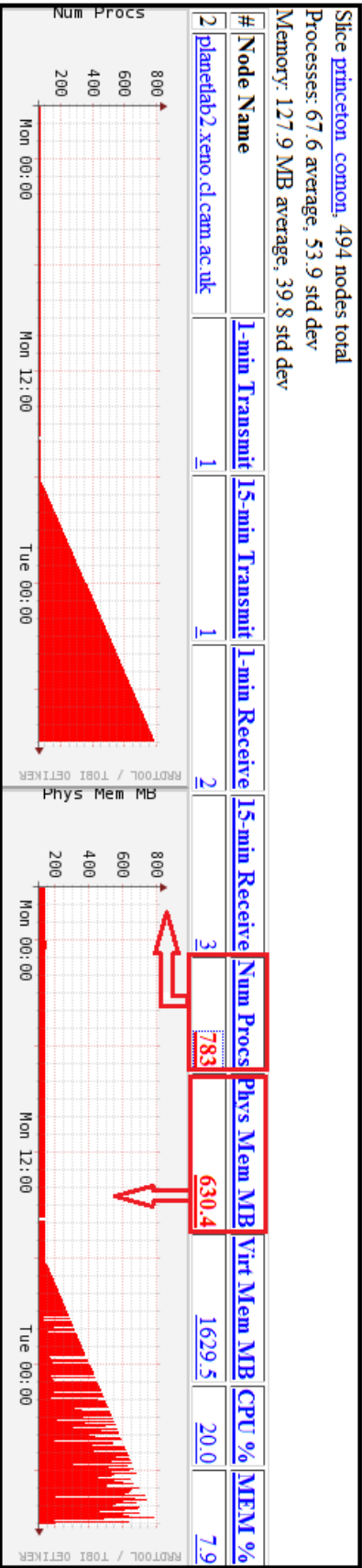
Obr. 10.2: Paralelné spúšťanie príkazov a terminálový výstup.



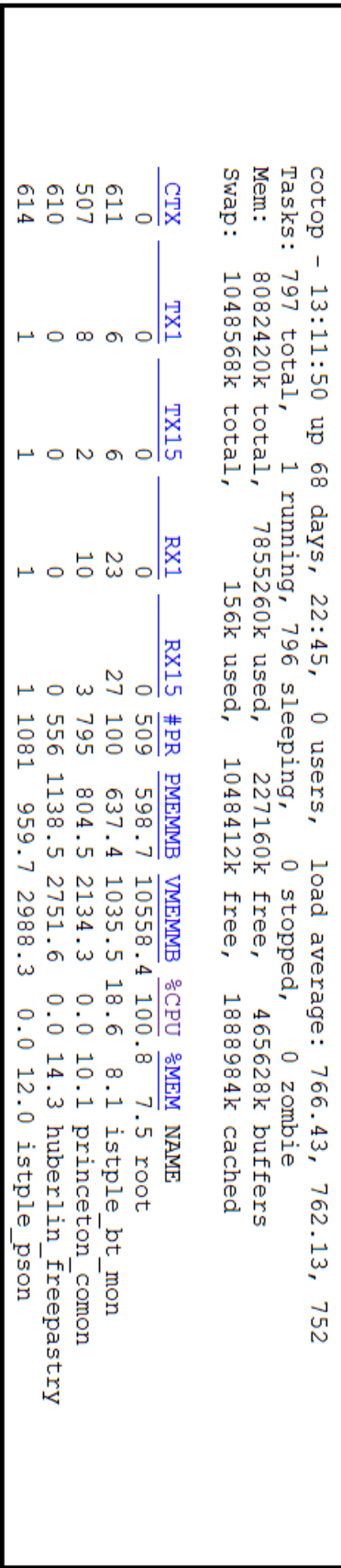
Obr. 10.3: Funkcia sťahovania a jej priebeh.

CoMon: zobrazenie priemerných nameraných hodnôt											
Slice Name	1-min Transmit	15-min Transmit	1-min Receive	15-min Receive	Num Procs	Phys Mem MB	Virt Mem MB	CPU %	MEM %	# Nodes	
princeton_comon	1.38	2.04	3.05	3.63	67.65	127.60	295.32	0.52	4.67	494	

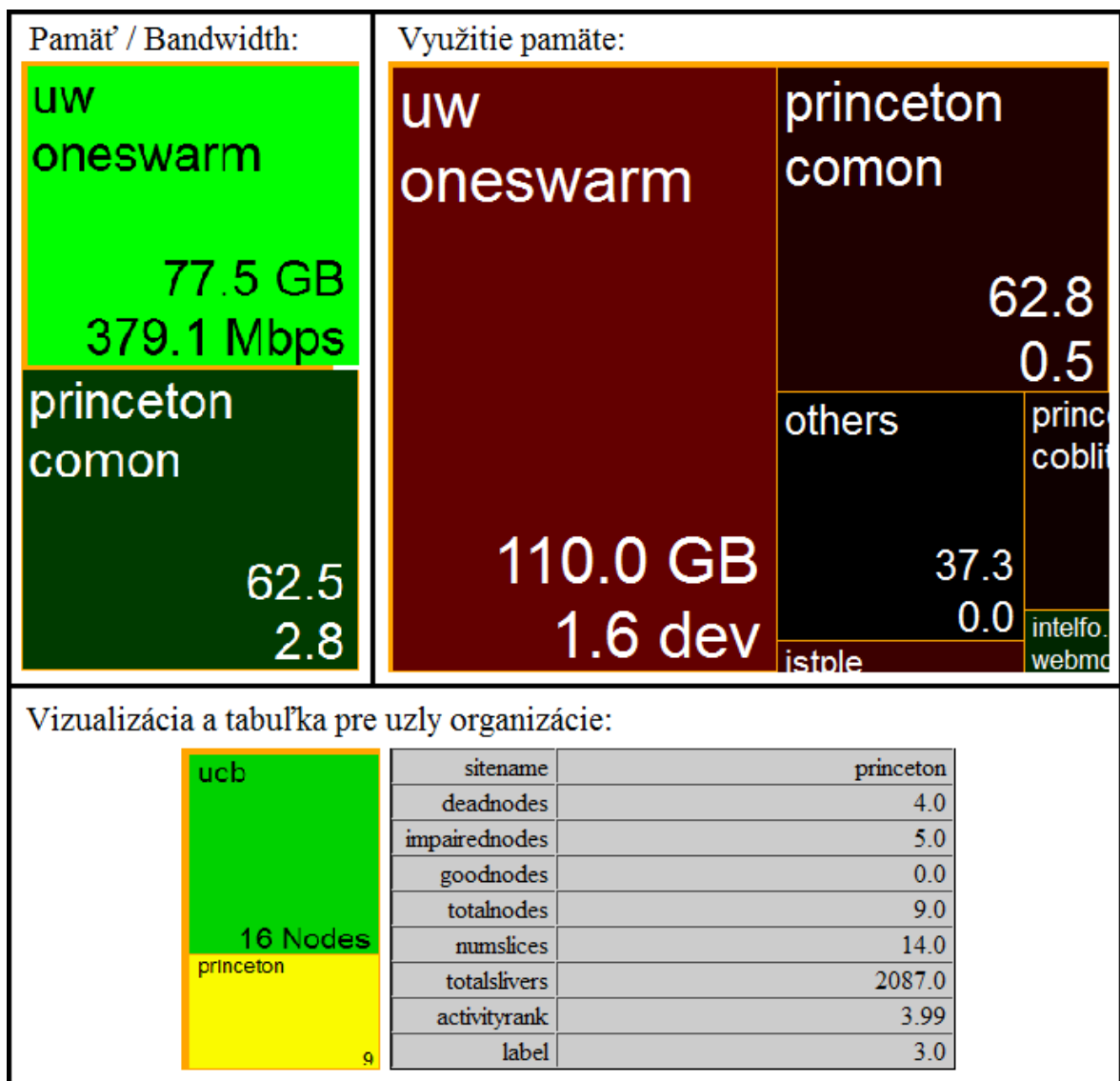
CoMon: zobrazenie problémových sliceov



CoTop: zoznam sliceov využívajúcich prostriedky uzlu



Obr. 11.1: Využitie monitorovacej služby CoMon a CoTop.



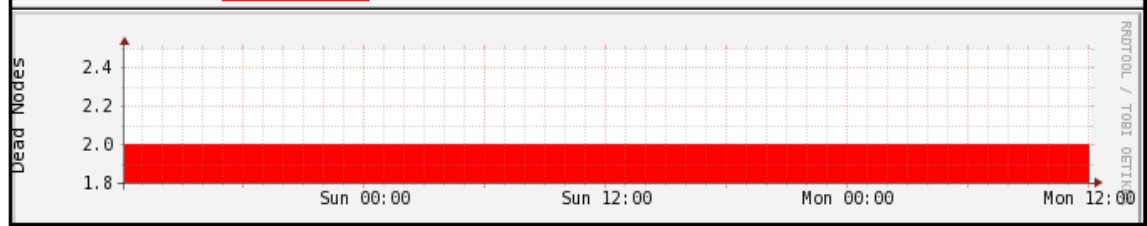
Obr. 11.2: Využitie modulu vizualizácií.

slicename	princeton_comon
sitehash	8.17095451E8
bwtot	2776.0
bwmax	51.0
bwavg	5.64
bwdev	2.55
bwnumdev	17.82
proctot	33259.0
procsmax	779.0
procsavg	67.6
procsdev	53.8
procsnumdev	13.22
phystot	62516.4
physmax	643.0
physavg	127.07
physdev	32.29
physnumdev	15.98
virttot	144401.8
virtmax	7202.8
virtavg	293.5
virtdev	401.05
virtnumdev	17.23
cputot	169.3
cpumax	9.0
cpuavg	0.34
cpudev	1.0
cpunumdev	8.64
bwwkmean	2822.14
bwwkdev	142.87
bwwknumdev	-0.32
memwkmean	62310.25
memwkdev	1118.79
memwknumdev	0.18
cpuwkmean	283.29
cpuwkdev	143.44
cpuwknumdev	-0.79

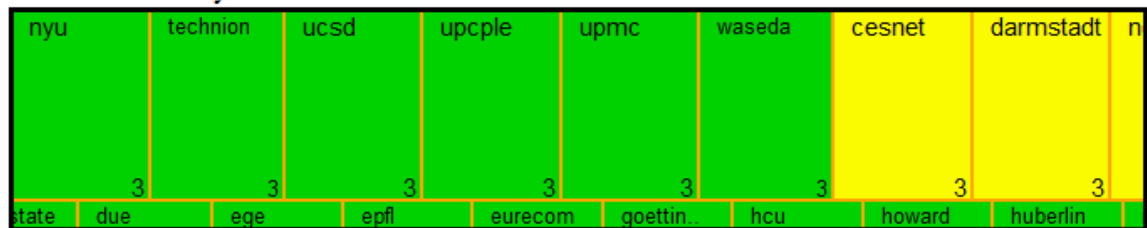
Tab. 11.1: Tabuľka hodnôt potrebných na zhotovenie vizualizácií.

HTML rozhranie CoMon: zoznam organizácií

#	Site Name	Dead Nodes	Impaired Nodes	Good Nodes	Total Nodes	Num Slices	Total Slivers
40	cesnet	2	0	1	3	6	13

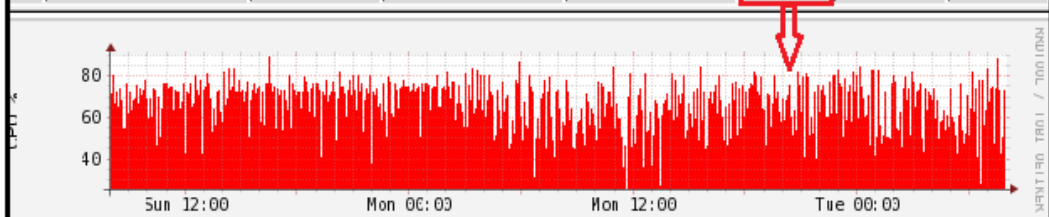


Vizualizácia využitia uzlov:



HTML rozhranie CoMon: zoznam sliceov s maximami nameraných hodnôt

#	Slice Name	Num Procs	Phys Mem MB	Virt Mem MB	CPU %	MEM %	# Nodes
32	cesnet_s2	17	13.0	40.8	59.0	0.6	6



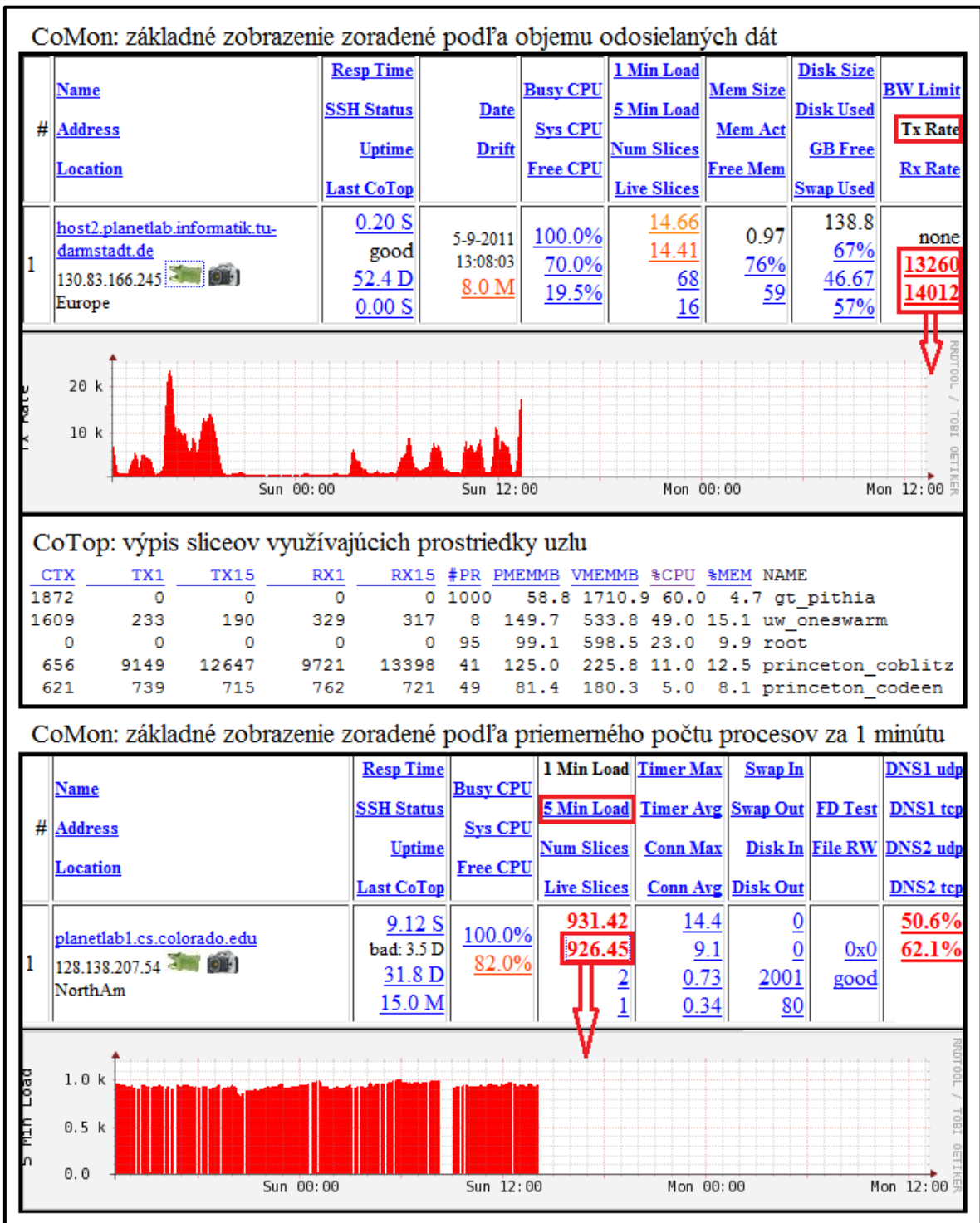
Výpis procesov z démoma služby CoTop na uzle pl2.csl.utoronto.ca:

#PR	PMEMMB	VMEMMB	%CPU	%MEM	NAME
14	13.0	33.7	74.0	0.4	cesnet_s2

Obr. 11.3: Využitie vizualizácie a monitorovacích služieb CoMon a CoTop.

#	Name Address Location	Resp Time SSH Status Uptime Last CoTop	Node Type Boot State Key OK	Date Drift	CPU Speed Num Cores Cores Per CPU	Busy CPU Sys CPU Free CPU	1 Min Load 5 Min Load Num Slices Live Slices	Timer Max Timer Avg Conn Max Conn Avg	Mem Size Mem Act Free Mem	Swap In Swap Out Disk In Disk Out	Disk Size Disk Used GB Free Swap Used	BW Limit Tx Rate Rx Rate
140	planetlab1.cesnet.cz 195.113.161.82  Europe	3.26 S good 31.3 D 0.00 S	Prod boot	5-9-2011 11:40:04 0.00 S	3.4 2 1	100.0% 23.0% 17.8%	19.70 20.43 41 13	37.8 11.0 1.23 0.34	1.97 63% 96	3 4 2524 198	261.3 6% 246.84 19%	10000 1432 1075
141	planetlab2.cesnet.cz 195.113.161.83  Europe	Connection refused good 26.1 D	Prod boot									none
142	pie1.cesnet.cz 195.113.161.13  Europe	0.22 S good 25.2 D 0.00 S	Prod boot	5-9-2011 11:42:01 0.00 S	2.8 4 4	34.0% 7.0% 99.2%	3.43 3.55 40 3	215.2 10.8 15.57 0.48	3.21 59% 100	0 13 1023	450.0 14% 389.64 2%	none 353 395
143	pie2.cesnet.cz 195.113.161.14  Europe	0.23 S good 25.2 D 0.00 S	Prod boot	5-9-2011 11:42:01 -0.00 S	2.8 4 4	81.0% 54.0% 67.0%	9.31 8.70 0 0	151.5 11.2 10000.00	3.21 60% 100	0 9 285	450.0 12% 400.12 2%	none 0 0
144	planetlab1.ft.vutbr.cz 147.229.10.250  Europe	Connection refused good 58.3 D	Prod boot									none

Obr. 11.4: Užij organizácie CESNET.



Obr. 11.5: Využitie služby CoMon na odhalenie možných problémov.

Name	CPU Hog		Tx Hog
Address	CPU %	Proc Hog	Tx Kb
Location	Mem Hog	Num Proc	Rx Hog
	Mem %		Rx Kb
planetlab02.cs.tcd.ie 134.226.52.35  Europe	google highground 29.0% google highground 73.3%	google highground 936	princeton codeen 151 princeton codeen 152
planetlab01.cs.tcd.ie 134.226.52.34  Europe	root 82.0% google highground 29.9%	google highground 403	nyu d 1237 nyu d 396
planet-lab1.ufabc.edu.br 200.133.215.141  SouthAm	root 46.0% colostate 557 28.9%	root 95	tsinghua mmlab 1682 tsinghua mmlab 1501

Obr. 11.6: Parametre rozšíreného zobrazenia služby CoMon.

Tab. 11.2: Tabuľka vytvorená zo zvolených dát vo formáte CSV.

Name	resptime [s]	cpuspeed [GHz]	freecpu [%]	1minload	numslices	memsize [GB]	memact [%]	gbfree [GB]	bwlimit [Kb]	txrate [Kbps]	rxrate [Kbps]
plgmu5.ite.gmu.edu	0,032	2,66	99,2	1,3	36	3,67	68,76	393,6	0	2025	1452
miranda.planetlab.cs.umd.edu	0,035	2,66	99	2,49	52	3,42	74,52	368,2	1	1867	1321
nodea.howard.edu	0,035	2,66	99,6	1,83	43	3,67	69,45	368,7	0	2160	1737
nodeb.howard.edu	0,037	2,66	99,2	0,39	39	3,67	72,25	390,7	0	178	672
pl1.cs.yale.edu	0,042	2,39	98	2,96	26	2,96	80,61	401,8	0	2749	2172
node-2.mcgillplanetlab.org	0,044	2,39	99	0,52	58	3,82	62,63	352,1	0	2503	1548
pl2.cs.yale.edu	0,044	2,39	99,6	0,96	22	3,95	67,76	438,6	0	2552	1671
plnode-03.gpolab.bbn.com	0,047	2,50	99,8	1,01	38	3,95	58,47	385,0	0	2042	2044
adras tea.mcs.suffolk.edu	0,049	2,49	99,6	1,45	53	7,92	58,48	586,6	10000	742	1465
plnode-04.gpolab.bbn.com	0,051	2,50	100	1,19	38	3,95	71,32	397,3	0	1646	1161
metis.mcs.suffolk.edu	0,055	2,49	99,8	4,44	50	7,92	56,01	595,2	10000	838	1261
pl2.rcc.uottawa.ca	0,061	2,67	99,6	0,78	0	7,93	46,60	608,9	0	0	0
planetlab-04.vt.nodes.planet-lab.org	0,068	2,66	99,8	1,77	37	3,67	72,06	372,9	0	43	118
planetlab-03.vt.nodes.planet-lab.org	0,069	2,66	99,8	0,19	32	3,67	65,64	425,9	0	2240	1768
planetlab2.clemson.edu	0,069	2,66	99,4	1,5	37	3,83	64,69	390,1	0	1891	1209
kc-sce-plab2.umkc.edu	0,090	2,66	99,4	0,49	41	3,67	62,28	398,3	1000	938	920