

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

MONITOROVÁNÍ SERVERŮ S OS LINUX

MONITORING OF SERVERS WITH OS LINUX

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Filip Šuba

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Dan Komosný, Ph.D.

BRNO 2018

Bakalářská práce

bakalářský studijní obor **Informační bezpečnost**

Ústav telekomunikací

Student: Filip Šuba

ID: 185944

Ročník: 3

Akademický rok: 2017/18

NÁZEV TÉMATU:

Monitorování serverů s OS Linux

POKYNY PRO VYPRACOVÁNÍ:

Popište možnosti získání informací o serverech s OS Linux. Zhotovte modul v jazyce Python 3, který bude získávat informace o aktuálním stavu serverů celosvětové sítě PlanetLab. K tomu využijte API funkce dostupné na www.planet-lab.org. Současnou aplikaci vyvíjenou na FEKT VUT v Brně pro vzdálenou práci se servery sítě PlanetLab převeďte do jazyka Python 3. Aplikaci dále upravte tak, aby mohl být připojen vámi vytvořený modul. Modul a upravenou aplikaci veřejně vystavte prostřednictvím repozitáře GitHub pod svobodnou licencí MIT. Zhotovte stručný popis celé aplikace a manuál pro její použití v anglickém jazyce.

DOPORUČENÁ LITERATURA:

[1] PUŽMANOVÁ, R. TCP/IP v kostce. 2. vyd. Kopp, 2009. 620 s. ISBN: 978-80-7232-388-3.

[2] PILGRIM, M. Ponořme se do Python(u) 3. CZ.NIC, 2010. 435 s. ISBN: 978-80-904248-2-1.

Termín zadání: 5.2.2018

Termín odevzdání: 29.5.2018

Vedoucí práce: doc. Ing. Dan Komosný, Ph.D.

Konzultant:

prof. Ing. Jiří Mišurec, 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

Táto bakalárska práca sa zaoberá monitorovaním serverov v rámci siete PlanetLab. V úvode je popísaná sieť PlanetLab, terminológia používaná v nej, ale aj projekty realizované vďaka nej. V rámci práce bol vytvorený modul v jazyku Python pre získavanie údajov o serveroch v rámci siete PlanetLab. Ďalej bola rozšírená aplikácia Plbmng, do ktorej bol integrovaný spomenutý modul. Aplikácia je uverejnená pod licenciou MIT na stránke www.github.com a taktiež ju je možné získať z indexu balíčkov pre jazyk Python www.pypi.org. K aplikácii bol vytvorený manuál a dokumentácia k modulu.

KĽÚČOVÉ SLOVÁ

PlanetLab, PlanetLab API, SSH, Linux, Python

ABSTRACT

This bachelor's thesis deals with monitoring of a servers in experimental network PlanetLab. The beginning of the thesis is about PlanetLab network, terminology used within it and also projects realized in it. For generating information about node, has been created a module in Python, which has been integrated into application Plbmng. Application has been published under the MIT license on www.github.com and on Python Package Index www.pypi.org. For the application has been created manual and for the module has been created documentation.

KEYWORDS

PlanetLab, PlanetLab API, SSH, Linux, Python

ŠUBA, Filip. *Monitorování serverů s OS Linux*. Brno, 2017, 45 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: doc. Ing. Dan Komosný, Ph.D.

VYHLÁSENIE

Vyhlasujem, že som svoju bakalársku prácu na tému „Monitorování serverů s OS Linux“ vypracoval(a) samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor(ka) uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil(a) autorské práva tretích osôb, najmä som nezasiahol(-la) nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý(-á) následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce pánovi doc. Ing. Danovi Komosnému, PhD. za odborné vedenie, konzultácie a podnetné návrhy k práci.

Brno

.....

podpis autora(-ky)

POĎAKOVANIE

Výzkum popsaný v tejto bakalárskej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pro inovace.

Brno

.....
podpis autora(-ky)

OBSAH

Úvod	9
1 Experimentálna sieť PlanetLab	10
1.1 Terminológia	10
1.2 Projekty realizované v sieti PlanetLab	11
1.3 PlanetLab API	12
1.4 PlanetLab data miner	13
2 Operačný systém Linux a virtualizácia	14
2.1 SSH – Secure Shell	16
2.2 Linux–VServer	17
3 Vytvorený modul a rozšírenie aplikácie Plbmng	18
3.1 Modul na získanie informácií o aktuálnom stave serverov	18
3.2 Výstup modulu	22
3.3 Použitie z príkazového riadku	24
3.4 Rýchlosť behu programu	25
3.5 Rozšírenie programu Plbmng	30
4 Dokumentácia	31
5 Distribúcia	33
6 Použitie a analýza výsledkov aplikácie Plbmng	35
7 Záver	39
Literatúra	40
Zoznam symbolov, veličín a skratiek	43
Zoznam príloh	44
A Obsah priloženého CD-R	45

ZOZNAM OBRÁZKOV

1.1	Hlavné menu dokumentácie zobrazené v prehliadači.	10
1.2	Hlavné menu programu Plbmng. Zdroj:[6]	13
3.1	Vývojový diagram modulu	19
3.2	Graf vyjadrujúci čas, za ktorý je možné získať informácie o 1000 ser- veroch s rôznymi postupmi	25
4.1	Hlavné menu dokumentácie zobrazené v internetovom prehliadači. . .	31
5.1	Stránka so zdrojovým kódom Plbmng na www.github.com	33
5.2	Stránka aplikácie Plbmng na PyPI.	34
6.1	Ukážka okna s informáciami o vybranom serveri.	37
6.2	Ukážka programu MC vzdialene pripojeného na server z obrázku 6.1. . .	37
6.3	Graf odpovedí serverov na ICMP správu.	38
6.4	Graf umiestnenia serverov na kontinentoch.	38

ZOZNAM VÝPISOV

3.1	Získanie všetkých uzlov.	20
3.2	Ukážka implementácie prekladu <i>hostname</i> na IP adresu	21
3.3	Ukážka výstupného súboru.	23
3.4	Príklad použitia modulu pomocou príkazového riadku.	24
3.5	Začiatok merania s postupom č. 1	26
3.6	Koniec merania s postupom č. 1	26
3.7	Začiatok merania s postupom č.2	27
3.8	Koniec merania s postupom č. 2	28
3.9	Získanie lokácie na základe zemepisných súradníc.	28
3.10	Výsledok merania postupom č. 3	29
4.1	Ukážka funkcie <i>main</i> zo skriptu po úprave	32
4.2	Ukážka vygenerovanej dokumentácie v čistom texte	32
6.1	Príkazy použité pri inštalácii <i>Plbmng</i>	35
6.2	Ukážka vyplneného konfiguračného súboru „ <i>plbmng.conf</i> “	36

ÚVOD

Účelom semestrálneho projektu je oboznámiť sa s experimentálnou sieťou PlanetLab a vytvorenie modulu v programovacom jazyku Python pre získanie informácií o serveroch dostupných v rámci siete PlanetLab. Pre získanie informácií je použité PlanetLab API, ktoré je vytvorené priamo pre manipuláciu so servermi. Získané informácie poskytujú základný prehľad o konfigurácii serveru, jeho lokácii a názvu inštitúcie, ktorá server prevádzkuje. Modul je vytvorený za účelom integrácie do aplikácie *plbmng* vytvorenej v rámci bakalárskej práce Ivana Andrašova [6].

Aplikácia *plbmng* je rozšírená a upravená tak, aby ju bolo možné zverejniť na www.pypi.org. K skriptom v jazyku Python je vytvorená dokumentácia podľa konvencie *PEP-257* [23]. K celej aplikácii je vytvorený stručný manuál s postupom ako aplikáciu nainštalovať a s popisom jej funkcií v anglickom jazyku. Zdrojový kód aplikácie je zverejnený na stránke www.github.com pod licenciou MIT.

Keďže aplikácia je vytvorená na použitie v operačnom systéme Linux a v rámci siete PlanetLab sa využíva virtualizácia, v kapitole 2 je popísaný daný operačný systém a jednoduchý úvod do virtualizácie. Pre vzdialené pripojenie k serveru sa používa protokol SSH, ktorý je popísaný v sekcii 2.1.

Kapitola 3 sa zaoberá vznikom modulu na získavanie serverov v rámci siete PlanetLab a rozšíreniu aplikácie *Plbmng*. Modul je možné importovať do skriptu, ale taktiež sa dá spustiť z príkazového riadku 3.3. Modul je vytvorený s ohľadom na jeho univerzálne použitie, resp. aby ho mohol použiť ktokoľvek v rámci siete PlanetLab. Na získanie údajov o lokácii serveru je použité reverzné geokódovanie 3.1. Štruktúra výstupného súboru z modulu a jeho ukážka je v sekcii 3.2. Rýchlosť modulu je analyzovaná v sekcii 3.4. Zmeny vykonané na rozšírenie aplikácie *Plbmng* sú spísané v sekcii 3.5.

V kapitole 4 je popísaný postup pri vytváraní dokumentácie a jej výsledný vzhľad. Dokumentácia k projektu je dostupná v prílohe.

Možnosti získania kópie aplikácie *Plbmng* sú popísané v kapitole 5.

V kapitole 6 je opísaný postup pri inštalácii aplikácie. Využitie aplikácie na vzdialené pripojenie na server ale aj analýza odpovedí serverov na ICMP správu.

1 EXPERIMENTÁLNA SIEŤ PLANETLAB

PlanetLab je svetová experimentálna sieť, ktorá podporuje vývoj nových internetových služieb, ktorá funguje od roku 2002. Pre svoj výskum ju použilo viac ako 1000 výskumníkov z univerzít, ale taktiež ju použili aj výskumníci z rôznych firiem zaoberajúcich sa výskumom distribuovaných systémov, mapovaní sietí, *peer-to-peer* systémov, distribuovanými hashovacími tabuľkami a spracovávaním dotazov. PlanetLab funguje ako konzorcium na čele s prestížnymi americkými univerzitami (University of California at Berkeley, Princeton University a University of Washington). Medzi členov konzorcia dnes patria aj veľké medzinárodné firmy ako Google, Intel alebo Hewlett Packard [1].

1.1 Terminológia

V rámci siete PlanetLab sú často spomínané pojmy, ktoré sú potrebné objasniť hneď na začiatku, pretože sú používané pri opise zloženia siete. V nasledujúcom zozname sú vybraté tie najdôležitejšie z [2]:

- **Node(uzol)** – node je dedikovaný server, na ktorom bežia časti služieb PlanetLab.
- **Site (miesto)** – fyzická lokácia, kde sa nachádzajú uzly.
- **Slice (diel)** – množina alokovaných fyzických prostriedkov v rámci pridelených uzlov. Po pridaní uzlu do dielu sa vytvorí virtuálny server, ku ktorému sa môže užívateľ pripojiť.
- **Sliver** – množina alokovaných prostriedkov na jednom uzle.
- **User(užívateľ)** – je ktokoľvek autorizovaný pre používanie siete.
- **Virtual Server (Virtuálny server)** – každý *sliver* je na uzle vytvorený pomocou VServer 2.2.



Obr. 1.1: Hlavné menu dokumentácie zobrazené v prehliadači.

1.2 Projekty realizované v sieti PlanetLab

Základnou myšlienkou siete PlanetLab je poskytovať zdroje pre testovanie aplikácií na rozvoj internetu. Dôležité je to, že pri testovaní svojich aplikácií pracujú vývojári priamo v podmienkach reálneho sveta a môžu testovať správanie sa danej aplikácie kdekoľvek na svete a v akýchkoľvek podmienkach. V súčasnej dobe je aktívnych okolo 600 výskumných projektov bežiacich v rámci siete PlanetLab.

Výber projektov, ktoré boli uskutočnené vďaka sieti PlanetLab:

- *Exploring BitTorrent peer distribution via hybrid PlanetLab-Internet measurement* [19] – projekt zaoberajúci sa pochopením peer-to-peer systémov a na základe toho vytvorenia nového peer-to-peer systému, ktorý bude viac vyhovovať poskytovateľom internetového pripojenia. Projekt taktiež opisuje prekonávanie výziev škálovateľnosti a reprezentatívnosti pri snahe sa pochopiť a merať peer-to-peer systémy v reálnych podmienkach. Opisuje sa v ňom aj nový spôsob merania, ktorý používa sieť PlanetLab ako platformu na interakciu s BitTorrent stopovačmi.
- *Delay measurement and analysis of network performance on PlanetLab* [20] – projekt sa zaoberá meraním výkonnosti internetovej siete a oneskorenia v rôznych lokalitách sveta. Tieto poznatky môžu hrať významnú rolu pri plánovaní sietí v lokalitách so zvýšenou záťažou siete.
- *Relying on randomness PlanetLab experiments with distributed file-sharing protocols* [21] – projekt sa venuje zdieľaniu súborov pomocou vylepšeného Chord algoritmu. Pri zdieľaní je súbor rozdelený na priemerne veľké súbory a sú následne ponúkané na stiahnutie. Rozsievač (seeder) a všetci partneri (peers), ktorí majú záujem o súbor sa náhodne medzi sebou kontaktujú. Využitie tohto spôsobu malo za následok značný nárast výkonu siete.
- *Finding stable cliques of PlanetLab nodes* [22] – v projekte sa venujú RTT (Round-Trip-Time) komunikácii medzi servermi v sieti PlanetLab, ktoré počítajú jej rozdiel. Na základe týchto údajov algoritmus určí či je server stabilný alebo nestabilný. Tento experiment poukázal na správanie serverov, s ktorým sa vopred nepočítalo.

1.3 PlanetLab API

PlanetLab API [4], bolo vytvorené pre účely manipulácie s centrálnou databázou PlanetLab. Pomocou tohto API je možné pridávať a upravovať všetky informácie užívateľoch, uzloch, miestach a všetkých celkovo dôležitých údajov potrebných viesť v databáze. API je využívané na internetovej stránke Planetlab, na serveroch a v rôznych skriptoch využívajúcich informácie z databázy.

Prvá a najdôležitejšia časť dokumentácie hovorí o spôsobe autentizácie voči API. Podporované typy sú:

- Autentizácia pomocou hesla – tento typ vyžaduje užívateľské meno a autentizačný reťazec. Užívateľské meno je zvyčajne e-mailová adresa a autentizačný reťazec heslo.
- Autentizácia pomocou GPG – užívateľ musí nahráť svoj verejný kľúč vopred pomocou preddefinovanej funkcie. Pre overenie musí užívateľ zadať užívateľské meno a špecifický podpis.
- Autentizácia pomocou ustanovenia relácie – tento typ nie je možné použiť samostatne. Najprv je nutné použiť jednu z predchádzajúcich možností autentizácie. Na vygenerovanie kľúča relácie sa používa funkcia *GetSession*. Kľúče relácie majú obmedzenú platnosť na 24 hodín.
- Anonymná autentizácia – užívateľ nezadáva žiadne informácie, ktoré by ho mohli identifikovať.

Po overení sa voči API, je možné začať používať funkcie dostupné v API. Prevažná väčšina funkcií začínajúca slovom *Get* podporuje filtrovanie. Ako filter môže byť použité nejaké identifikačné číslo, štruktúra s podmienkami alebo reťazec. Taktiež aby sme nemuseli dostať všetky návratové hodnoty, ktoré funkcia podporuje, vieme si definovať vopred o aké hodnoty máme záujem. Príklad použitia filtrov vo vytvorenom module sú dostupné vo výpise 3.1.

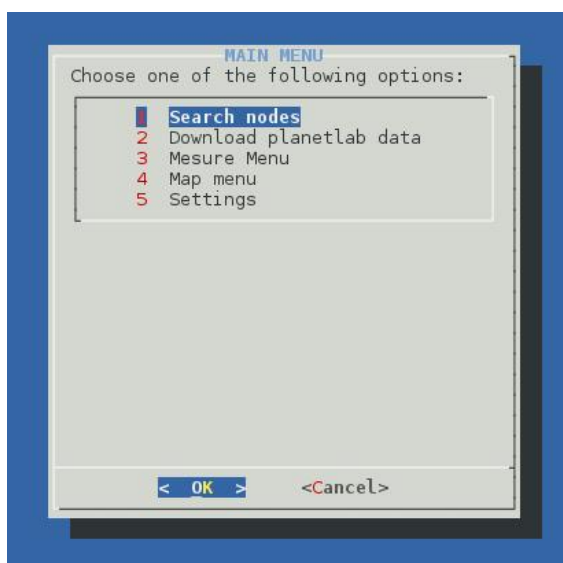
Funkcie z API sú rozdelené do troch rozhraní:

- Rozhranie registra – v tomto rozhraní sú definované funkcie na pridanie záznamu alebo získanie už vyplneného záznamu. Z tohto rozhrania sú použité ďalej v práci funkcie *GetNodes*, *GetSites*, *GetInterfaces* [5].
- Rozhranie manažmentu – funkcie pod týmto rozhraním sú určené na konfigurovanie, spravovanie uzlov a taktiež na získanie ich aktuálneho stavu.
- Rozhranie dielov – funkcie rozhrania poskytujúce diely/slivery a kontrolu nad nimi.

V každom rozhraní sa funkcie delia na verejné a privátne. K privátnym má prístup vždy len administrátor.

1.4 PlanetLab data miner

S týmto názvom je označené programové riešenie z bakalárskej práce Ivana Andrašova [6], ktoré slúži ako nástroj pre správu serverov v rámci siete PlanetLab. Program je kombináciou dvoch programovacích jazykov, a to konkrétne Python a Bash. Hlavné menu je zobrazené na obrázku 1.4. Funkcie tohoto programu sú testovanie aktuálnej dostupnosti serveru pomocou ICMP správy, možnosť vzdialeného pripojenia pomocou protokolu SSH alebo pomocou programu s názvom *Midnight Commander*, automatizácia merania dostupnosti pomocou ICMP správ, ale aj dostupnosti pripojenia pomocou SSH, filtrovanie a zobrazovanie údajov serverov na základe lokality, DNS názvu alebo pomocou IP adresy. Program pracuje s údajmi o serveroch, ktoré sú dostupné v súbore s názvom *planetlab.node*. Súbor musí mať predom určenú štruktúru pretože program predpokladá, že na presne stanovenom mieste v riadku sa bude nachádzať určitá informácia. Ak by totiž nebola dodržaná, nebudeme schopný využiť žiadny z nástrojov v tomto programe. Na základe tejto podmienky som staval svoj modul v jazyku Python, ktorý nahradil predošlý. Predošlý modul pracoval priamo s HTML kódom webovej stránky, čo ho značne limitovalo. Po zmene dizajnu stránky prestal správne fungovať a bolo potrebné pristúpiť ku zmene prístupu riešenia tohoto problému.



Obr. 1.2: Hlavné menu programu Plbmng. Zdroj:[6]

2 OPERAČNÝ SYSTÉM LINUX A VIRTUALIZÁCIA

Aplikácia Plbmng je vytvorená pre použitie v operačnom systéme Linux, a preto je dôležité si ho predstaviť. Operačný systém Linux začal vznikať na začiatku 90. rokov 20. storočia ako podobná verzia operačného systému Unix. Linus Torvalds chcel vytvoriť operačný systém, ktorý bude bezplatný a plne kompatibilný s operačným systémom Unix. Ako sa uvádza na [3] v roku 1991 vytvoril prvé Linux jadro a v roku 1992 sa stalo toto jadro bezplatným softvérom.

V súčasnosti sa dá spustiť takmer na hocijakom zariadení, od mikrokontrolérov až po superpočítače. Najčastejšie sa môžeme stretnúť s jadrom Linux na serveroch. Taktiež existujú Linuxové distribúcie, ktoré priamo obsahujú balíčky s grafickým užívateľským rozhraním, aby sa čo najviac prispôbili používaniu na stolnom počítači. Nedá mi nespomenúť, že Linux aj po toľkých rokoch stále zostáva bezplatný a určuje trend, ktorým by sa mohli vydať aj ostatné platformy v budúcnosti.

V tejto práci sa hlavne budem zaoberať správou operačného systému Linux pomocou príkazového riadku. Príkazový riadok sa podľa môjho názoru dá definovať ako rozhranie, vďaka ktorému som schopný spúšťať príkazy. Ako je známe každá operácia, ktorú počítač vykonáva, nám odoberá zdroje, ktoré môžeme využiť efektívnejšie. Keďže spracovávanie reťazcov je pre dnešné zariadenia jednoduchšie ako vykresľovanie užívateľského rozhrania, je logické použiť na správu zariadenia príkazový riadok. Podobne môžeme rozmýšľať pri ovládaní vzdialeného zariadenia pri komunikácii pomocou siete internet. Pri odoslaní príkazu, čo je vlastne reťazec znakov v počítačovej reči, vyjadríme ako niekoľko bajtov. Pár bajtov nepredstavuje pre dnešnú rozsiahlu sieť internet žiaden problém. V dnešnej dobe sa už dá prenášať cez internet celé užívateľské rozhranie v reálnom čase. V minulosti nebol možný prenos veľkého objemu dát z dôvodu slabej infraštruktúry. Preto bola niekedy jediná voľba príkazový riadok. Najpoužívanejší nástroj na správu vzdialeného zariadenia je v dnešnej dobe stále SSH, ktorému sa budem venovať bližšie v nasledujúcej sekcii 2.1.

Na serveroch, s ktorými som pracoval v rámci siete PlanetLab som používal linuxové distribúcie CentOS a Fedora. Zaradil by som ich medzi tie najznámejšie. Nedá mi nespomenúť taktiež distribúcie Ubuntu a Debian, ktoré majú tiež veľkú užívateľskú základňu. Linux, je len jadro operačného systému a aby bolo použiteľné je potrebné k nemu nainštalovať rôzne balíčky. Balíček je nejaké rozšírenie, ktoré nám pridáva určité funkcie ako napríklad ovládače grafickej karty. Všetky distribúcie už obsahujú balíčky, ktoré určujú hlavné zameranie takejto distribúcie. Najčastejšie sa môžeme stretnúť s distribúciami zameranými na server alebo pracovnú stanicu. Vo

väčšine prípadov rozlišuje takéto distribúcie to, že či daná distribúcia obsahuje grafické užívateľské rozhranie. Ako som spomínal vyššie, na serveroch sa dnes využíva len vo výnimočných prípadoch. Naopak pri pracovných staniciach je dnes k dispozícii veľa grafických užívateľských rozhraní a užívateľ si môže vyberať. Niektoré sa snažia vzhľadom čo najviac priblížiť súčasným najpoužívanejším operačným systémom Windows a MacOS. Iné sú jedinečné a snažia sa vytvoriť si svoju identitu a nájsť si svoje publikum na základe toho. Teraz by som sa chcel trochu bližšie venovať distribúciám Fedora a CentOS, keďže som na nich strávil množstvo času.

Ako sa môžeme dozvedieť z [9], Fedora je operačný systém tvorený komunitou ľudí, ktorí sa snažia vytvoriť platformu, ktorá je úplne zadarmo. Dôležité je spomenúť, že Fedora je platforma zložená len zo softvéru, ktorý má otvorený zdrojový kód a teda je úplne zadarmo. To znamená, že to ako funguje, je všetko dostupné z internetu a každý človek sa môže pozrieť na zdrojový kód a ak mu nejaká funkcia chýba, môže ju vytvoriť a ak prejde určitým procesom, môže sa stať súčasťou distribúcie. Taktiež ak niekto nájde chybu, môže ju nahlásiť alebo rovno opraviť. Takto môže ktokolvek prispieť do zlepšenia Fedory. Toto neplatí len pre Fedoru, ale aj pre všetky linuxové distribúcie. Pre porovnanie takéto niečo nie je možné pri operačnom systéme Windows, kde nemáme k dispozícii zdrojový kód a tým pádom ani nevidíme čo všetko môže nejaká funkcia vykonávať alebo sa tam môže nachádzať závažná chyba a nikto o tom nemusí vedieť.

Ďalšia distribúcia, s ktorou som pracoval je CentOS. CentOS je odvodený od Red Hat Enterprise Linux a taktiež je s ním plne kompatibilný. Je vyvíjaný komunitou a jeho hlavným cieľom je byť stabilným operačným systémom zameraným hlavne na cloud, vedecké spracovávanie dát a servery. Tak ako skoro každá distribúcia Linux je úplne zadarmo a je voľne šíriteľný [10].

Okrem toho, že tieto dve distribúcie sú založené na jadre Linux majú spoločné to, že obidve distribúcie sú späté z s Red Hat Enterprise Linux a firmou Red Hat ako takou. Spomínam to preto, že je dôležité pochopiť ako aj vďaka tomuto spojeniu sa dokážu distribúcie rýchlejšie rozširovať. V distribúcii Red Hat Enterprise Linux sa zameriava hlavne na firemnú sféru a snaží sa podporovať to, čo si platiaci zákazníci želajú. Za bezplatnú verziu podobnú RHEL môžeme považovať distribúciu CentOS, ktorá je derivát od RHEL. V CentOS sa vyskytujú balíčky pre *cloud*, ale taktiež pre server, na ktorom bežia rôzne aplikácie. Fedora bude lepšia aj vďaka tomu, že najnovšie inovácie, ktoré sa budú vyskytovať v RHEL, sa taktiež budú nachádzať aj v nej a vďaka tomu bude komunita stále môcť zasahovať do vývoja najnovších technológií pre Linux [11].

2.1 SSH – Secure Shell

Pripojenie na server je realizované v aplikácii pomocou protokolu SSH. SSH protokol nám slúži na bezpečné vzdialené prihlásenie medzi klientom a serverom. Pod pojmom bezpečnú si môžeme predstaviť to, že po nadviazaní a ustanovení spojenia sú všetky odosielané príkazy šifrované a tým pádom ukryté. SSH protokol je založený na štruktúre klient-server. Pre ustanovenia spojenia sa používa asymetrická kryptografia. Klient iniciuje komunikáciu a na základe verejného kľúča servera si overí jeho identitu. Keď je server overený, tak sa nadviaže komunikácia, ktorá je šifrovaná symetrickou kryptografiou. Komunikácia prebieha pomocou protokolu TCP na transportnej vrstve, nad ktorou sa nachádza vrstva troch najdôležitejších súčastí architektúry SSH protokolu. Tieto tri súčasti su nasledovné:

- **Protokol transportnej vrstvy** – na tejto vrstve protokolu SSH sa overuje identita serveru, integrita spojenia a že či dané spojenie je dôveryhodné
- **Protokol overenia užívateľa** – táto vrstva slúži na overenie užívateľa, ktorý sa chce pripojiť na SSH server. Táto vrstva je nad transportnou vrstvou protokolu SSH.
- **Protokol spojenia** – úlohou tohto protokolu je rozdeliť jeden šifrovaný tunel do niekoľko virtuálnych kanálov. Tento protokol je najvyššie vo vrstve protokolu SSH.

Po prijatí žiadosti od klienta na nadviazanie bezpečného spojenia sa najprv ustanoví pomocou protokolu transportnej vrstvy spojenie a následne sa odosiela žiadosť o službu. Po dokončení autentifikácie pomocou protokolu overenia užívateľa sa pošle druhá žiadosť o službu. Ako posledný sa použije protokol spojenia, ktorý poskytuje množstvo kanálov na použitie pre základne funkcie ako nadviazanie bezpečnej relácie pre shell a tunelovanie vlastných portov protokolu TCP a spojenia X11, ale taktiež dané kanály sa môžu použiť na široký záber rôznych iných funkcií [13].

Nie len príkazy sa dajú odosielať pomocou protokolu SSH, ale protokol SFTP opisuje aj prenos súborov. SFTP sa už nezaobrá nadviazaním spojenia, ale využíva už vytvorené. V operačnom systéme Linux využíva tento protokol program scp, ktorý je možné používať z príkazového riadku. SFTP protokol pracuje na princípe výzva–odpoveď a každá takáto výzva má svoje číslo, z čoho vyplýva, že môže byť súčasne spustených viac prenosov naraz [14].

2.2 Linux–VServer

VServer je nástroj pre vytvorenie niekoľkých virtuálnych serverov na jednom fyzickom zariadení. V tejto sekcii sa budem venovať zo začiatku všeobecne virtualizácii, ale spomeniem aj konkrétne fungovanie softvéru VServer.

Keďže výkon počítačov rastie tak, je možné tento výkon rozdeliť medzi niekoľko užívateľov. Každé virtuálne zariadenie je nezávislé od ostatných a môže využívať len časť pridelených zdrojov. Dnes sa na trhu vyskytuje množstvo druhov softvéru, ktoré umožňujú vytvárať virtuálne zariadenia. Rozlišujú sa napríklad druhom virtualizácie alebo úrovňou abstrakcie. Poznáme veľa druhov virtualizácie, tu spomeniem aspoň zopár:

- **Virtualizácia na úrovni operačného systému** – znamená že na jednom jadre operačného systému môže bežať niekoľko virtuálnych serverov, ktoré sa tvária ako samostatné zariadenie. Pri tomto type virtualizácie nie je možné spustiť rôzne operačné systémy.
- **Prirodzená virtualizácia** – operačný systém, ktorý chceme spustiť pomocou prirodzenej virtualizácie, musí byť kompatibilný s hardvérom, pretože hypervisor len oddeľuje jednotlivé stroje.
- **Emulácia** – softvér umožňuje, aby bolo možné spustiť operačný systém, ktorý nie je kompatibilný s hardvérom. Tento softvér sa snaží simulovať všetky stavy, ktoré by ten náš s dostupným hardvérom nedokázal.

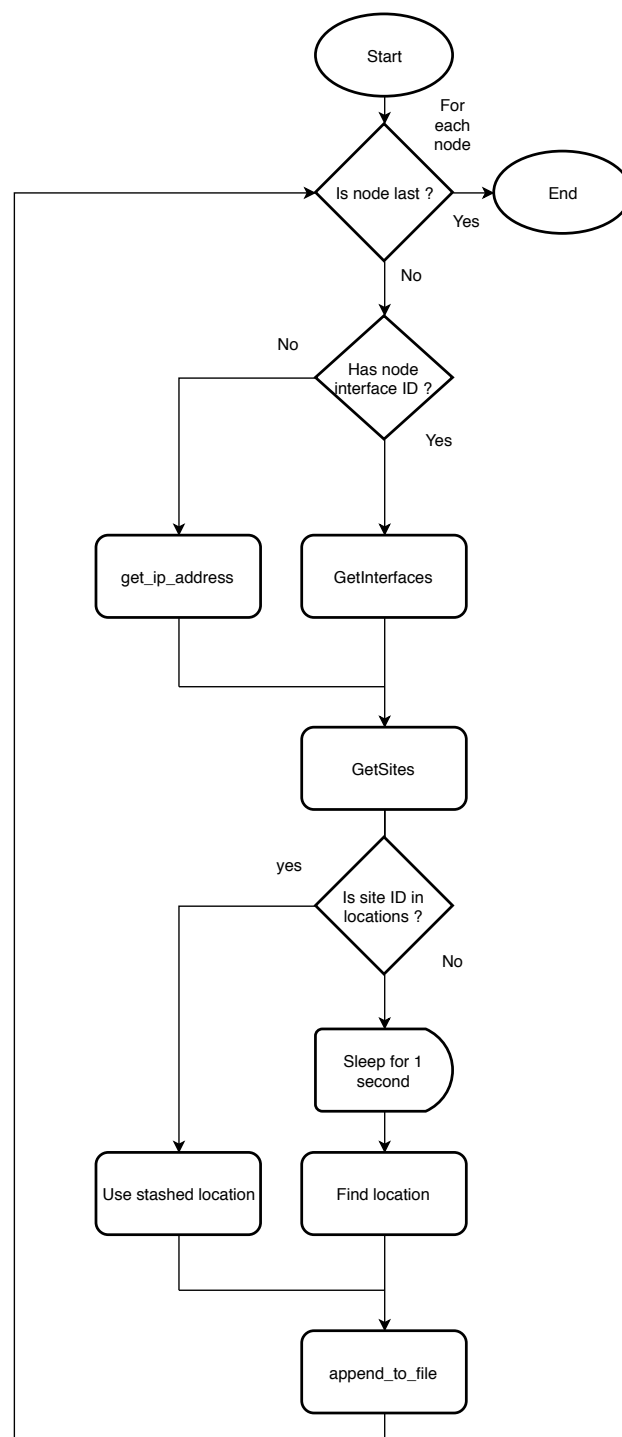
Linux–VServer je softvér, ktorý vykonáva virtualizáciu na úrovni operačného systému. Jadro operačného systému je prostredník medzi aplikáciami a hardvérom. Vďaka tomuto je možný tento typ virtualizácie. Servery väčšinou majú väčší výpočetný výkon ako je potrebné na beh jedného operačného systému a preto aj vďaka VServeru je možné využiť aj ten nevyužívaný blok výpočetného výkonu. Na každom virtuálnom servere bežia všetky základné služby a taktiež každý má samostatného užívateľa root. Toto je vďaka VServeru umožnené bez takmer akýchkoľvek dodatočných zásahov. Týmto serverom nie sú určené žiadne hranice využitia zdrojov, čiže ak virtuálny server vyžaduje ďalšie jadro CPU pre nejakú službu a CPU má jadro, ktoré nie je aktuálne využívané, tak mu ho prideliť. Toto môžeme považovať za veľmi efektívny prístup k virtualizácii. Vďaka tomu, že mnoho distribúcií nevyužíva špeciálne upravené jadro Linux, je možné spustiť na viacerých virtuálnych strojoch rôzne distribúcie [15].

3 VYTVORENÝ MODUL A ROZŠÍRENIE AP- LIKÁCIE PLBMNG

3.1 Modul na získanie informácií o aktuálnom stave serverov

Základnú komunikáciu opisuje dokumentácia PlanetLab API dostupná z [4]. Toto API nie je určené na import priamo do skriptu, ale je potrebné komunikovať so vzdialeným serverom, na ktorom sa nachádza. K takémuto účelu nám slúži knižnica `xmlrpclib`, ktorá umožňuje volanie funkcií umiestnených na vzdialenom serveri. Ako argument je potrebné použiť URL adresu serveru, kde sa dané API nachádza. Tak ako sa píše v sekcii 1.3, pre autentizáciu sa môže využiť viacero spôsobov. Pre modul som zvolil autentizáciu pomocou prihlasovacieho mena a hesla. Pri volaní nejakej funkcie z API je vždy potrebné predať jej ako argument slovník so spôsobom autentizácie, prihlasovacím menom a heslom. Ako ďalšie argumenty, ktoré funkcia vyžaduje je potrebné predať filter a návratové informácie. Toto je základná schéma každej volanej funkcie, ktorá je určená pre návrat informácií. Vývojový diagram programového riešenia sa nachádza na obrázku 3.1.

Modul som navrhol tak, aby bol funkčný ako samostatná jednotka, ktorú je možné používať z príkazového riadku. Fungovanie modulu týmto spôsobom je vysvetlené neskôr v sekcii 3.3. Pri použití modulu v programe `Plbmng` je modul zavolaný z príkazového riadku a sú mu predané potrebné argumenty pre správnu činnosť. Tento program som preložil do jazyku Python 3, kde je použitie modulu odlišné a popis jeho fungovania je vysvetlený taktiež neskôr v kapitole. Rozdiel medzi použitím z príkazového riadku a využitím programu napísaného v jazyku Python je len v predaní argumentov. Princíp na získanie údajov je rovnaký. Po zadaní prihlasovacích údajov sa odošle požiadavka pomocou `xmlrpclib` na server, od ktorého žiadam aby mi vrátil späť informácie o všetkých serveroch zoradené podľa identifikačného čísla. Tieto informácie mi majú pomôcť dopátrať sa ku všetkým potrebným informáciám potrebným k správe serverov. Štruktúra získaná zo serveru je tzv. slovník. V jazyku Python táto štruktúra priradzuje hodnotu k stanovenému kľúčovému výrazu. Funkcia `GetNodes` z API mi vráti takýto slovník s informáciami o identifikačnom čísle miesta, kde sa server nachádza, ďalej identifikačné číslo rozhrania serveru a názov hostiteľa (*hostname*). Ukážka implementácie v skripte sa nachádza vo výpise 3.1.



Obr. 3.1: Vývojový diagram modulu

```

import xmlrpclib

plc_host = "www.planet-lab.eu"

auth = {'AuthMethod': 'password',
        'Username': 'studentutko@gmail.com',
        'AuthString': '*****'}

api_url = "https://%s:443/PLCAPI/" % plc_host
plc_api = xmlrpclib.ServerProxy(api_url, allow_none=True)

all_nodes = plc_api.GetNodes(auth, {"-SORT": "node_id"},
                               ["node_id", "site_id", "hostname", "interface_ids"])

```

Výpis 3.1: Získanie všetkých uzlov.

Tieto identifikačné čísla nemajú informačnú hodnotu vysokú ako také. Dôvod prečo ale nie sú zanedbateľné je ten, že ich vyžadujú funkcie, ktoré využívam ne-skôr. Identifikačné číslo rozhrania je predané funkcii s názvom *GetInterfaces*. Tento názov môžeme voľne preložiť ako získaj rozhrania, a teda z názvu vyplýva podstata funkcie. Ako parameter jej predávam identifikačné číslo rozhrania a ako návratovú hodnotu požaduje IP adresu tohto rozhrania. Po získaní prvotných výsledkov sa mi naskytl pohľad na malé množstvo vrátených IP adries. Keďže tento údaj je jeden z najdôležitejších ak nie najdôležitejší, pretože bez tejto informácie nie sme schopní spravovať server vzdialene. Nástroje používané na vzdialenú správu serveru bez tejto informácie nedokážu pracovať. Na základe týchto dostupných informácií a prečítaní dokumentácie k API som zvolil prístup k získaniu IP adresy pomocou prekladu názvu hostiteľa. Na tento postup som zvolil vstavanú knižnicu jazyku Python s názvom *Socket*. Funkcia z tejto knižnice je schopná zistiť z názvu hostiteľa IP adresu. Táto možnosť získania IP adresy je využitá len v prípade nevrátenia tejto hodnoty z funkcie *GetInterfaces* a jej účelom je teda len zvýšiť množstvo získaných adries. Pri testovaní funkcií z API, ktoré mali vracať mesto a krajinu kde sa server nachádza, sa mi len v minime prípadov podarilo zistiť tieto údaje. Preto som pomocou funkcie *GetAddress* získal zemepisnú šírku a zemepisnú dĺžku. Podrobnejšie je opísaná práca s týmito údajmi v odseku 3.1. Po nazbieraní všetkých potrebných údajov z API, sú údaje exportované do súboru s názvom *default.node*. Jeho štruktúra je navrhnutá presne podľa požiadaviek programu Plbmng 3.1.

```

def get_ip_address(hostname):
    """
    Function translates hostname to IP address.\n
    :param hostname: HOSTNAME\n
    :return: if hostname cannot be translated to IP
            address returns None. Otherwise it returns IP
            address as a string.
    """
    if hostname is not None or hostname is not '':
        try:
            ip_addr = socket.gethostbyname(hostname)
            return ip_addr
        except socket.error:
            pass
    return None

```

Výpis 3.2: Ukážka implementácie prekladu *hostname* na IP adresu

Reverzné geokódovanie

Na základe informácií o lokalizácii serverov, ktoré som spomenul v sekcii 3.1 som zvolil metódu s názvom reverzné geokódovanie. Podstata tejto metódy je, že dokáže preložiť zemepisnú dĺžku a šírku na presný bod na mape. Poskytovatelia geolokačných služieb majú túto metódu implementovanú vo svojich API a sú schopní poskytnúť všetky dostupné informácie o danom mieste. Väčšina poskytovateľov poskytuje túto službu za poplatok alebo umožňuje len obmedzený počet prekladov za deň. Ja som sa rozhodol použiť poskytovateľa geolokačných služieb s názvom *Open Street Maps*. Po preklade tohoto názvu nám už prvé slovo hovorí čo je hlavným zmyslom tohto projektu. Slovo otvorené má evokovať, že mapy sú doslova otvorené. Každý môže prispieť do projektu vylepšením mapy svojho okolia. Aj podmienky použitia [7] majú čo najmenej limitovať osobu, ktorá ich chce použiť vo svojom projekte. Podmienku, ktorú som ja sám musel implementovať do svojho modulu je obmedzenie na jednu požiadavku o preklad za 1 sekundu. Pre získanie týchto údajov, pripravených pre ďalšie použitie v jazyku Python, som využil knižnicu s názvom *Geocoder* [16]. Knižnica je univerzálna a dokáže pracovať aj s inými poskytovateľmi ako len *Open Street Maps*. Vo výstupnom súbore som použil región, mesto a kód krajiny, ktorý som využil taktiež aj pre získanie kontinentu. Kontinent som určil pomocou slovníka, kde ako kľúč je kód krajiny a hodnota je vždy kód kontinentu.

3.2 Výstup modulu

Všetko čo bolo zatiaľ spomenuté o programovom riešení vedie k tomuto bodu. Mojm cieľom bolo spraviť modul čo najviac nezávislí od programu Plbmng. Preto sú implementované dva formáty výstupu.

Prvý typ výstupu je hlavne pre použitie už so spomenutým programom. Údaje sú exportované ako súbor s názvom *default.node*, jeho ukážka je zobrazená vo výpise 3.3. V podstate je to textový súbor so špecifickou príponou pre jednoduchšiu identifikáciu. Štruktúra súboru je daná tým, že na každom riadku sú vždy len informácie o jednom serveri. Medzi každou informáciou je tabulátor, ktorý funguje pri spracovaní súboru ako označenie pre program, kde končí a kde začína ďalšia informácia. V tabuľke 3.1 sú vypísané všetky informácie, ktoré sa nakoniec v súbore nachádzajú. V prípade, že nejakú informáciu nebolo možné získať pomocou vyššie spomenutých funkcií modulu, sa objaví vo výstupnom súbore slovo *unknown*, čo v preklade znamená neznámy.

Druhá možnosť ako vrátiť zozbierané informácie je pomocou parametru *return_output* funkcie *run*. Tento parameter zabezpečí návrat informácii v štruktúre jazyku Python s názvom *list*. Funkcia v základnom nastavení má zakázanú túto možnosť, ale predpokladám, že ak by niekto použil túto funkciu vo svojom programe, nastavenie jedného parametru je lepšie ako pri každom zavolaní funkcie vrátiť všetky tieto informácie aj, keď sú nevyžiadané.

Názov	Popis
ID	ID uzlu
IP	IP adresa uzlu
DNS	DNS meno uzlu
continent	kontinent, kde je uzol umiestnený
country	krajina, kde je uzol umiestnený
region	región, kde je uzol umiestnený
city	mesto, kde je uzol umiestnený
URL	URL inštitúcie, ktorá uzol spravuje
full name	celý názov inštitúcie, ktorá uzol spravuje
latitude	zemepisná šírka
longitude	zemepisná dĺžka

Tab. 3.1: Formát výstupného súboru.

#	ID	IP	DNS	continent	country	region	city	url	
			full_name	latitude	longitude				
03		131.130.69.164	ple02.fc.univie.ac.at	EU	at	unknown	unknown	http://www.cs.univie.ac.at	University of Vienna 48.16 16.24
04		221.199.217.144	planetlab1.research.nicta.com.au	OC	au	Sydney	Sydney	http://www.nicta.com.au	National ICT Australia (NICTA) -33.8954 151.197
05		193.157.115.250	planetlab1.ifi.uio.no	EU	no	Oslo	Oslo	http://www.ifi.uio.no/	University of Oslo 59.93 10.75
06		141.24.249.129	pl1.prakinf.tu-ilmenau.de	EU	de	Ilm	-Kreis	unknown	www.tu-ilmenau.de Technische Universitat Ilmenau 50.681 10.9386
07		130.192.9.230	planetlab1.polito.it	EU	it	TO	Torino	http://www.tlc-networks.polito.it	Politecnico di Torino 45.07 7.67
08		193.191.148.249	planck249ple.test.iminds.be	EU	be	Gent	Gent	http://www.ibbt.be	IBBT 51.04 3.74
09		193.136.19.29	planetlab-um10.di.uminho.pt	EU	pt	Braga	unknown	http://cctc.di.uminho.pt	CCTC / Universidade do Minho 41.5615 -8.39722
010		193.167.187.185	planetlab3.hiit.fi	EU	fi	Helsingin	seutukunta	http://www.hiit.fi	Helsinki Institute for Information Technology 60.1 25.0
011		185.38.3.76	planetlab1.virtues.fi	EU	fi	Oulun	seutukunta	http://www.cwc oulu.fi/home/	Centre for Wireless Communications 65.0582 25.4713
012		unknown	pl1.ccsr.fi.net	EU	gb	Surrey	unknown	http://www.ee.surrey.ac.uk/CCSR/	University of Surrey 51.2436 -0.588343
013		193.136.166.68	planetlab-2.tagus.ist.utl.pt	EU	pt	OER	unknown	http://www.ist.utl.pt	Instituto Superior Tecnico 38.7373 -9.30324

Výpis 3.3: Ukážka výstupného súboru.

3.3 Použitie z príkazového riadku

Naprogramovanie tejto možnosti použitia modulu bolo za podmienok integrácie do programu Plbmng nevyhnutné. Vo výpise 3.4 sú zobrazené informácie ako používať a aké možnosti použitia sú dostupné. Argumenty, bez ktorých nie je možné modul využívať sú *USERNAME*, *PASSWORD* a *OUTPUT*. *USERNAME* je prihlasovacie meno na internetovú stránku siete PlanetLab a *PASSWORD* je heslo. Tieto argumenty sú nevyhnutné z dôvodu vzdialeného pripojenia k API. *OUTPUT* je cesta k súboru kde sa majú informácie uložiť. Súbor nemusí byť vytvorený vopred, modul je schopný ho vytvoriť za užívateľa. V prípade zadanej neplatnej adresy je program ukončený s chybovým hlásením. *START_ID* slúži na určení začiatkovej hodnoty, od ktorej budú nájdené servery číslované. *QUIET* indikuje modulu, či si užívateľ praje zobrazovať informácie o serveri ako štandardný výstup. Ak užívateľ nešpecifikuje tento argument modul bude vypisovať informácie, v opačnom prípade sa vypíše len čas, kedy modul začal svoju činnosť.

```
usage: planetlab_list_creator [-h] -u USERNAME -p
    PASSWORD -o OUTPUT [--start_id START_ID] [-q]

Plb list creator is a script, which will return file with
    all available nodes from www.planet-lab.eu.
Required arguments: USERNAME, PASSWORD and OUTPUT.

optional arguments:
-h, --help show this help message and exit
-u USERNAME, --username USERNAME
-p PASSWORD, --password PASSWORD
-o OUTPUT, --output OUTPUT Path where you want to save
    output.
--start_id START_ID    ID for the first node
-q, --quiet             Suppress stdout
```

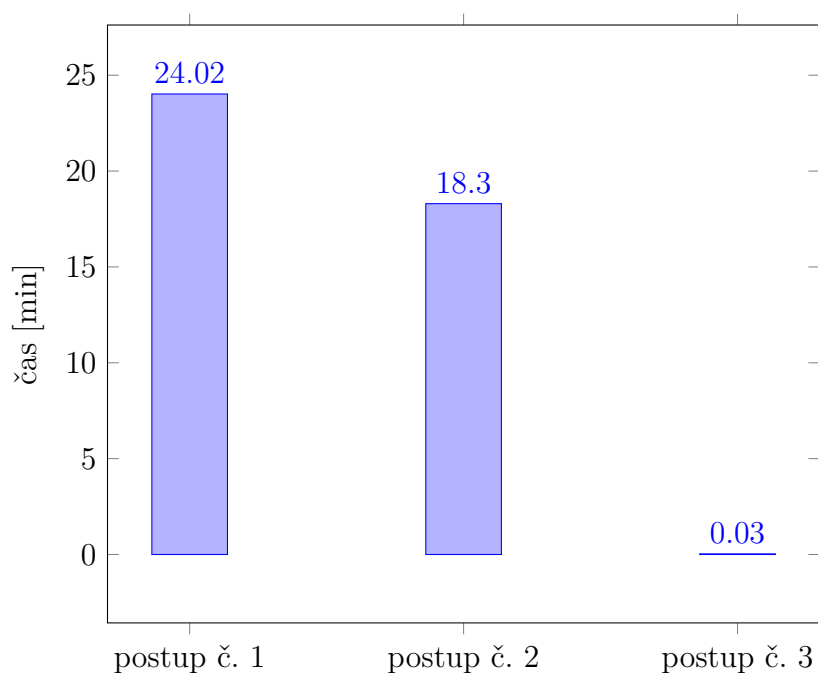
Výpis 3.4: Príklad použitia modulu pomocou príkazového riadku.

3.4 Rýchlosť behu programu

V tejto sekcii rozoberiem časové hladisko modulu. Ako počiatočný bod tejto analýzy budeme uvažovať dĺžku trvania získania informácií o 1000 serverov. Ako sa spomína v 3.1 program je pozastavený na jednu sekundu z dôvodu podmienok použitia [7]. Táto skutočnosť značne ovplyvňuje čas, za ktorý je schopný program dobehnúť do konca. Na obrázku 3.4 sa nachádza graf, kde je vyobrazený čas v minútach pri 3 rôznych postupoch. Merania boli prevedené v nadväznosti, aby sa zamedzilo skreslenie výsledkov zmenou počtu serverov.

Postup č. 1

Postup v grafe s číslom 1 je prvá verzia modulu. Modul v tejto verzii získal 1000 serverov a pre každý jeden, ktorý mal k dispozícii zemepisnú dĺžku a šírku spustil reverzné geokódovanie z *OpenStreetMaps*. Ukážka kódu s touto metódou je vo výpise 3.9. Modul sa ukončil po 24 minútach 2 sekundách, čo je relatívne dosť dlhá doba. Na obrázku a je zobrazený testovací beh modulu s časovým údajom o začiatku a serverom s ID 1 a koncom so serverom s ID 1000.



Obr. 3.2: Graf vyjadrujúci čas, za ktorý je možné získať informácie o 1000 serveroch s rôznymi postupmi

```
Start: Wed May 16 18:34:05 2018
ID: 1
IP ADDRESS: 212.235.189.115
HOSTNAME: planetlab2.fri.uni-lj.si
CONTINENT: EU
COUNTRY: SI
REGION: Upravna Enota Ljubljana
CITY: unknown
URL: http://www.uni-lj.si
NAME: University of Ljubljana
LATITUDE: 46.0319
LONGITUDE: 14.3052
```

Výpis 3.5: Začiatok merania s postupom č. 1

```
ID: 1000
IP ADDRESS: 139.30.241.192
HOSTNAME: pl2.uni-rostock.de
CONTINENT: EU
COUNTRY: DE
REGION: Rostock
CITY: Rostock
URL: http://www.uni-rostock.de
NAME: University of Rostock
LATITUDE: 54.13
LONGITUDE: 12.145

Finish: Wed May 16 18:58:06 2018
```

Výpis 3.6: Koniec merania s postupm č. 1

Postup č. 2

Na základe výsledkov som potreboval nejakým spôsobom skrátiť túto dobu a ako reakcia na to vznikla druhá verzia modulu, kde som si vytvoril databázu so získanými informáciami o miestach, kde sa servery nachádzajú. Toto bolo možné sprevádzkovať na základe toho, že server mal informáciu, kde sa nachádza označenú ako identifikačné číslo miesta. Preto je možné predpokladať, že dva servery majúce rovnaké identifikačné číslo miesta, sa nachádzajú na tom istom mieste. Preto som si vždy

uložil identifikačné číslo a k nemu som si priradil informácie, ktoré by bolo potrebné získať z *OpenStreetMaps*. Čiže v priebehu modulu ak sa ešte identifikačné číslo nenachádza v mojej databáze, tak sa odošle žiadosť na geokódovanie súradníc a potom sa uložia tieto informácie k identifikačnému číslu. Ak narazím na server s identifikačným číslom z databázy, modul sa nepozastavuje na 1 sekundu ako pri potrebe geokódovania, ale priamo sa k nemu priradia už predtým získané hodnoty. Ako je možné vidieť v grafe 3.4, ušetrilo to takmer 6 minút.

```
Start: Wed May 16 18:07:43 2018
ID: 1
IP ADDRESS: 212.235.189.115
HOSTNAME: planetlab2.fri.uni-lj.si
CONTINENT: EU
COUNTRY: SI
REGION: Upravna Enota Ljubljana
CITY: unknown
URL: http://www.uni-lj.si
NAME: University of Ljubljana
LATITUDE: 46.0319
LONGITUDE: 14.3052
```

Výpis 3.7: Začiatok merania s postupom č.2

Postup č.3

Predchádzajúce postupy podľa môjho názoru vyjadrujú skôr teoretické podmienky pri používaní modulu. Na testovanie tohto modulu som si na internetovej stránke siete PlanetLab pridal do mojej skupiny serverov všetky, ktoré boli v tom čase dostupné. Bolo to z dôvodu testovania najzložitejších podmienok, ktoré môžu nastať. Postup č. 3 v grafe 3.4 vyjadruje čas, za ktorý je možné získať informácie o jednom mieste. Takže ak budeme uvažovať, že užívateľ využíva len servery z niekoľkých lokácií, dĺžka behu programu sa bude pohybovať v sekundách a nie už v minútach.

Ďalšia možnosť ako by sa dal znížiť čas behu programu, je využitie menej restriktívnej organizácie pri využívaní jej služieb. Pri využití platených služieb som nenašiel žiadne časové obmedzenie žiadostí, ale pre zachovanie bezplatnosti využitia tohto modulu som sa snažil použiť nejakú bezplatnú službu.

```
ID: 1000
IP ADDRESS: 139.30.241.192
HOSTNAME: pl2.uni-rostock.de
CONTINENT: EU
COUNTRY: DE
REGION: Rostock
CITY: Rostock
URL: http://www.uni-rostock.de
NAME: University of Rostock
LATITUDE: 54.13
LONGITUDE: 12.145
```

```
Finish: Wed May 16 18:22:16 2018
```

Výpis 3.8: Koniec merania s postupom č. 2

```
sleep(1) 1
if node["latitude"] is not None and node["longitude"] is 2
    not None:
        g = geocoder.osm([node["latitude"], node["longitude"] 3
            ], method="reverse")
        if g.city is not None: 4
            node["city"] = g.city 5
        if g.county is not None: 6
            node["region"] = g.county 7
        if g.country_code is not None: 8
            node["country"] = g.country_code.upper() 9
            node["continent"] = get_continent(g.country_code) 10
```

Výpis 3.9: Získanie lokácie na základe zemepisných súradníc.

```
Start: Wed May 16 18:30:17 2018
ID: 1
IP ADDRESS: 195.113.161.84
HOSTNAME: planetlab3.cesnet.cz
CONTINENT: EU
COUNTRY: CZ
REGION: okres Hlavní město Praha
CITY: Praha
URL: http://www.ces.net/about/
NAME: CESNET - Czech Education and Research
    Network
LATITUDE: 50.102
LONGITUDE: 14.3916
ID: 2
IP ADDRESS: 195.113.161.83
HOSTNAME: planetlab2.cesnet.cz
CONTINENT: EU
COUNTRY: CZ
REGION: okres Hlavní město Praha
CITY: Praha
URL: http://www.ces.net/about/
NAME: CESNET - Czech Education and Research
    Network
LATITUDE: 50.102
LONGITUDE: 14.3916

ID: 3
IP ADDRESS: 195.113.161.82
HOSTNAME: planetlab1.cesnet.cz
CONTINENT: EU
COUNTRY: CZ
REGION: okres Hlavní město Praha
CITY: Praha
URL: http://www.ces.net/about/
NAME: CESNET - Czech Education and Research
    Network
LATITUDE: 50.102
LONGITUDE: 14.3916
Finish: Wed May 16 18:30:19 2018
```

Výpis 3.10: Výsledok merania postupom č. 3

3.5 Rozšírenie programu Plbmng

Pri práci na module som sa snažil aj o vylepšenie funkcií základného programu. Program Plbmng je uverejnený na stránke *GitHub*, ktorý som si skopíroval a pracoval na ňom bez ohrozenia jeho základnej funkcionality. Niektoré zmeny sa týkali len premenovania možností v menu kvôli zlepšeniu prehľadnosti a orientácie v ňom. Medzi dôležité zmeny, ktoré boli vykonané, patrí ukladanie užívateľských informácií. V predchádzajúcej verzii sa meno, heslo, názov a cesta ku privátnemu kľúču ukladala priamo do programu ako premenná. Ja som použil radšej iný prístup ako ukladať tieto dôležité údaje. Vytvoril som konfiguračný súbor s názvom *plbmng.conf* 6.2, ktorý sa nachádza v priečinku *bin*. Štruktúra súboru je jednoduchá, na každom riadku musí byť vždy len jedna informácia. Názov premennej musí byť vždy na ľavej strane od „=“ a hodnota na pravej. Ukladať tieto informácie do súboru je možné priamo upravovaním súboru alebo použiť možnosť *Settings* v hlavnom menu aplikácie.

Možnosť *Scrap planetlab.eu* bola presunutá z hlavného menu do *Measure menu* a bola premenovaná na *Monitor now*. Táto možnosť spúšťa modul spomenutý vyššie v texte. Modulu sú predané prihlasovacie meno a heslo z konfiguračného súboru. Cesta, kde bude súbor uložený je určená príkazom, ktorý vráti cestu k aktuálnej pracovnej zložke. Pred spustením sa zobrazí okno s varovaním, že získavanie údajov môže trvať približne 20 minút. Najčastejšie sa mi pri testovaní podarilo zadať zlé meno a heslo preto, ak sa modulu nepodarí pripojiť k API a ukončí sa, zobrazí sa okno s upozornením, že užívateľ pravdepodobne zadal zlé meno alebo heslo.

V menu *Set monitoring period* je pridaná možnosť *Run measurement now*, ktorá umožňuje okamžité spustenie monitorovania elementov, ktoré si užívateľ špecifikoval v *Measure menu*. Novo pridaná možnosť využíva skript, ktorý bol vytvorený pre nastavenie periodického monitorovania. S tým rozdielom, že je spustený okamžite ako štandardný skript a nie je pridaný do programu s názvom *Cron*, ktorý zabezpečuje spustenie skriptu v stanovenej perióde.

V *Node menu* je pridaná informácia o dostupnosti vzdialeného pripojenia pomocou SSH. Dostupnosť je testovaná v momente, keď je generované menu s informáciami. V tomto menu bol ešte doplnený časový údaj, kedy bola otestovaná dostupnosť serveru na ICMP správu.

V Python skriptoch generujúcich mapy zo zemepisnej šírky a dĺžky bolo potrebné upraviť predanie týchto hodnôt zo súboru. Modul na vykreslenie máp potrebuje predať súradnice ako desatinné číslo. Predtým toto nebolo ošetrované a preto, som pridal do kódu nech sa skript pokúsi zmeniť reťazec na desatinné číslo. V prípade, že reťazec nie je číslo, skript vyhodí chybu.

4 DOKUMENTÁCIA

Skripty v jazyku Python sú zdokumentované v prílohe A. V tejto prílohe sú popísané jednotlivé funkcie v štandarde, používanom pri dokumentácii jazyka Python [23]. Príloha obsahuje exportovanú dokumentáciu ako stránku v jazyku HTML, ktorú je možné zobrazíť po spustení súboru *index.html* v akomkoľvek prehliadači internetových stránok. Menu je interaktívne, čiže po kliknutí užívateľa na určitý modul sa otvorí daná stránka s popisom všetkých funkcií, ktoré obsahuje. Taktiež je možné využiť možnosť vyhľadávania v dokumentácii. Aby bolo možné vygenerovať dokumentáciu, bolo potrebné upraviť skripty na vytvorenie máp. Celý kód generujúci tieto mapy som zabalil do funkcie *main*. Pod túto funkciu som umiestnil podmienku, že keď sa Python pokúsi spustiť skript, spustí sa funkcia *main*. Na začiatok som umiestnil popis funkcie skriptu ako tzv. *docstring*. Ukážka týchto zmien sa nachádza vo výpise 4.1. Nástroje na generovanie dokumentácie k projektu vezmú len túto časť a použijú ju ako popis funkcie. Na vygenerovanie dokumentácie som použil nástroj Sphinx [24]. Sphinx umožňuje vytvárať dokumentáciu pre celú škálu jazykov a podporuje niekoľko variantov exportovania. Ja som svoju dokumentáciu vygeneroval, ako je už spomenuté, v jazyku HTML 4 a k tomu ešte aj ako čistý text do textových súborov 4.2.

The screenshot shows a web browser displaying the Sphinx-generated documentation for a project named 'plbmng'. The page has a two-column layout. The left column contains a 'Table Of Contents' with links to 'Welcome to plbmng's documentation!', 'Indices and tables', 'This Page', 'Show Source', and a 'Quick search' box with a 'Go' button. The main content area on the right features a large heading 'Welcome to plbmng's documentation!' followed by a bulleted list of links: 'plbmng' (selected), 'Description', 'Dependencies', 'Instalation', and 'Basic usage'. Below this is another heading 'Indices and tables' followed by a list of links: 'Index', 'Module Index', 'Search Page', 'planetlab_list_creator module', 'ssh_map module', 'map_one module', 'full_map module', and 'icmp_map module'. At the bottom of the page, a footer line reads: '©2018, Filip Suba. | Powered by Sphinx 1.7.1 & Alabaster 0.7.10 | Page source'.

Table Of Contents

[Welcome to plbmng's documentation!](#)
[Indices and tables](#)

This Page

[Show Source](#)

Quick search

Welcome to plbmng's documentation!

- [plbmng](#)
 - [Description](#)
 - [Dependencies](#)
 - [Instalation](#)
 - [Basic usage](#)

Indices and tables ¶

- [Index](#)
- [Module Index](#)
- [Search Page](#)
- [planetlab_list_creator module](#)
- [ssh_map module](#)
- [map_one module](#)
- [full_map module](#)
- [icmp_map module](#)

©2018, Filip Suba. | Powered by [Sphinx 1.7.1](#) & [Alabaster 0.7.10](#) | [Page source](#)

Obr. 4.1: Hlavné menu dokumentácie zobrazené v internetovom prehliadači.

```
def main():
    """
    Creates map with location of n node.\n
    :return: map_one.html file.
    """
    latitude = float(sys.argv[1])
    :
if __name__ == "__main__":
    main()
```

Výpis 4.1: Ukážka funkcie main zo skriptu po úprave

```
map_one module
*****

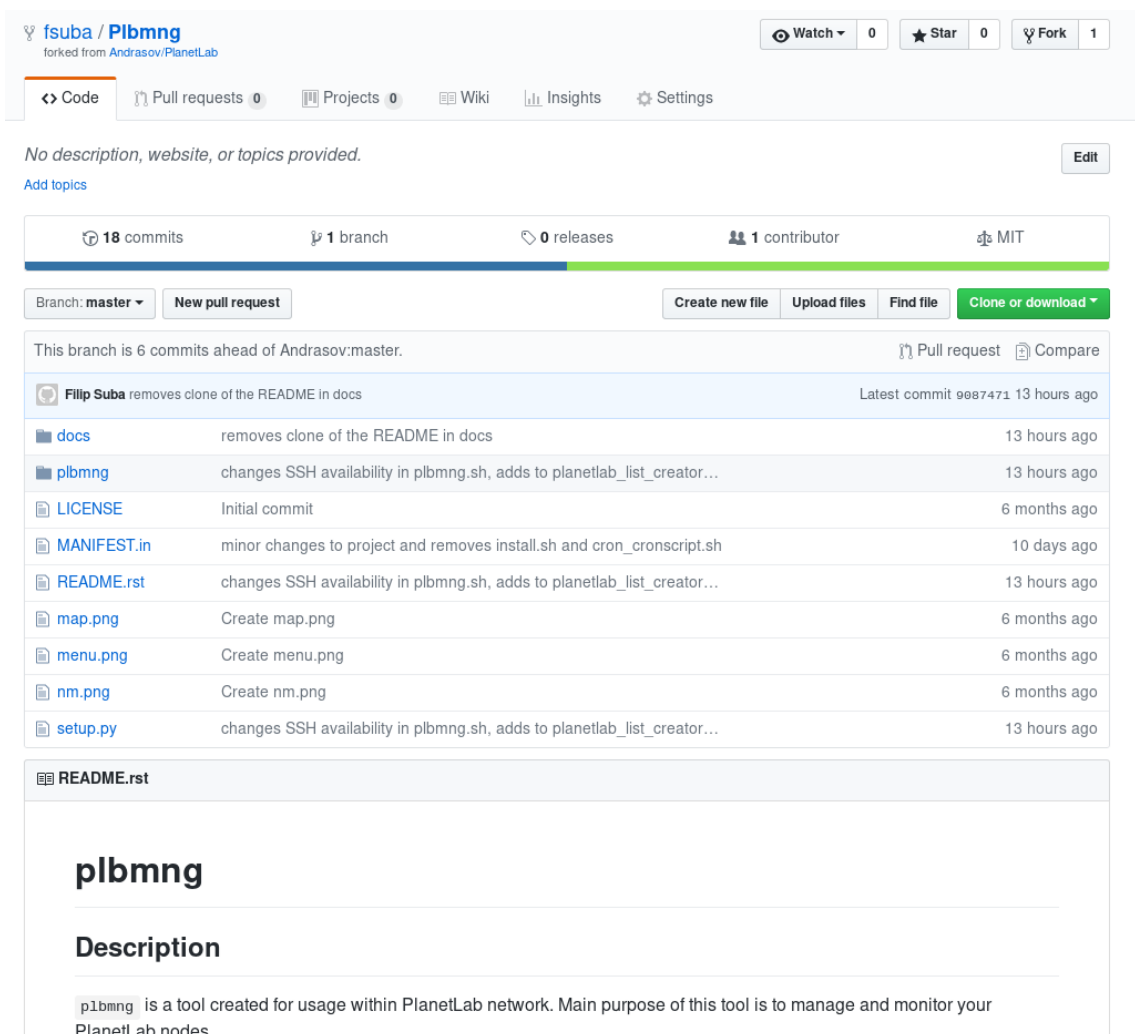
map_one.main()
    Creates a map with location of a node.
    Returns:
        map_one.html file
```

Výpis 4.2: Ukážka vygenerovanej dokumentácie v čistom texte

Dôležitou súčasťou dokumentácie je súbor *README*, ktorý je priložený v prílohe A. V úvode je uvedený účel a vymenovanie funkcií aplikácie. Ďalej sú tam vymenované balíky v jazyku Python, ktoré užívateľ musí mať nainštalované, aby mu aplikácia fungovala správne. V prípade, že užívateľ inštaluje aplikáciu z PyPI, balíky sa mu nainštalujú automaticky. Toto je zabezpečené v súbore „setup.py“, ktorý sa spúšťa pri každej inštalácii z PyPI. Ak užívateľ využije možnosť získania aplikácie stiahnutím zo stránky *GitHub*, musí si sám nainštalovať tieto balíky z PyPI alebo manuálne spustiť „setup.py“. V závislostiach sú uvedené aj programy, ktoré je nutné mať nainštalované vo svojej distribúcii operačného systému Linux. Tento proces nie je automatizovaný ako pri Python balíkoch a zostáva na každom užívateľovi. Všetky príkazy potrebné pre inštaláciu sú uvedené vo výpise 6.1. Ďalej sa v súbore *README* nachádza stručný popis možností a orientácie v menu aplikácii. Tento súbor je napísaný v reštrukturalizovanom texte, čo mi umožnilo vytvoriť prehľadný a dobre čitateľný návod.

5 DISTRIBÚCIA

Kópiu aplikácie je možné získať dvomi spôsobmi. Pri vývoji aplikácie som používal systém kontroly verzií s názvom Git. Ako hostiteľa som využil službu GitHub, čo je v súčasnosti jeden z najpoužívanejších poskytovateľov tejto služby [17]. Sú tam dostupné všetky zmeny, ktoré som postupom času pri vývoji aplikácie vykonal. Aplikácia je licencovaná licenciou MIT [18]. Táto licencia je jedna z najmenej obmedzujúcich využitie alebo rozšírenie softvéru a je často využívaná pri akademických projektoch. Jediná podmienka je, že každá kópia softvéru musí mať priložené znenie tejto licencie. Kópia z tejto stránky je dostupná v rôznych kompresných formátoch.



fsuba / **Plbmng**
forked from Andrasov/PlanetLab

Watch 0 Star 0 Fork 1

Code Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided. [Add topics](#) [Edit](#)

18 commits 1 branch 0 releases 1 contributor MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

This branch is 6 commits ahead of Andrasov:master. [Pull request](#) [Compare](#)

Commit	Message	Time
Filip Suba	removes clone of the README in docs	Latest commit 9e87471 13 hours ago
docs	removes clone of the README in docs	13 hours ago
plbmng	changes SSH availability in plbmng.sh, adds to planetlab_list_creator...	13 hours ago
LICENSE	Initial commit	6 months ago
MANIFEST.in	minor changes to project and removes install.sh and cron_cronscript.sh	10 days ago
README.rst	changes SSH availability in plbmng.sh, adds to planetlab_list_creator...	13 hours ago
map.png	Create map.png	6 months ago
menu.png	Create menu.png	6 months ago
nm.png	Create nm.png	6 months ago
setup.py	changes SSH availability in plbmng.sh, adds to planetlab_list_creator...	13 hours ago

README.rst

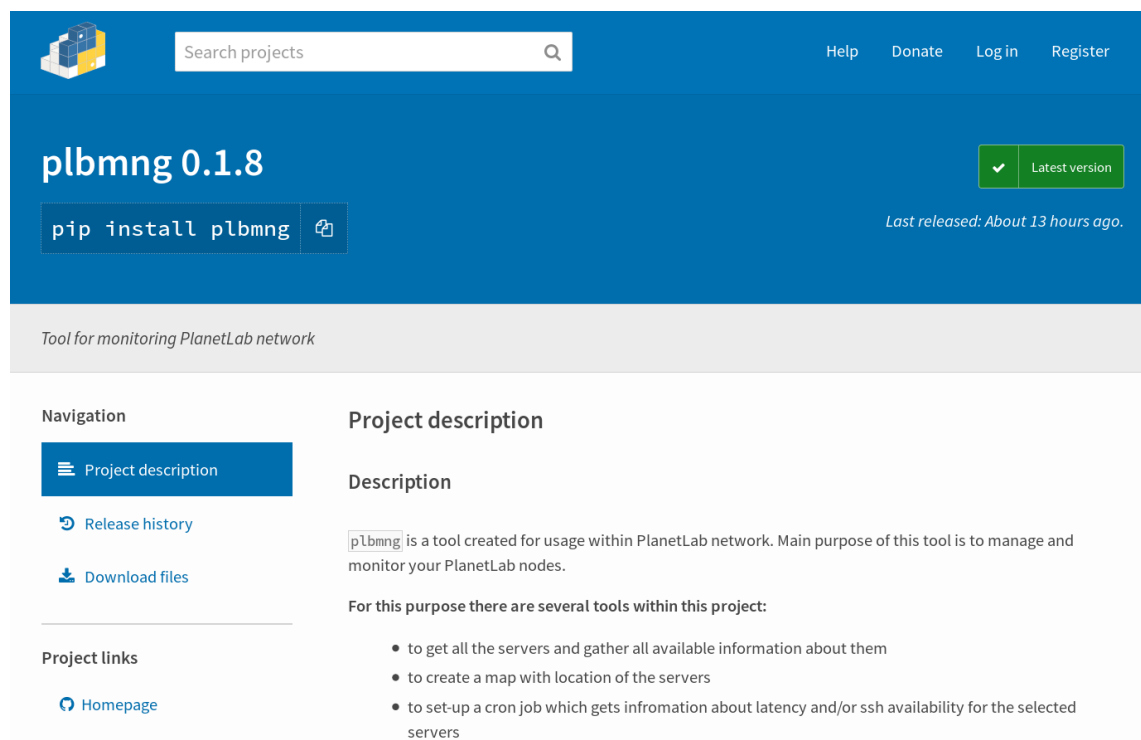
plbmng

Description

plbmng is a tool created for usage within PlanetLab network. Main purpose of this tool is to manage and monitor your PlanetLab nodes.

Obr. 5.1: Stránka so zdrojovým kódom Plbmng na www.github.com.

Ako druhá možnosť získania kópie aplikácie je stiahnuť ju pomocou programu pip z *Python Package Indexer*, skrátene PyPI. Na PyPI je možné nahrať svoj projekt, aby bol ľahko dostupný a mohol mať k nemu prístup ktokoľvek kto by ho potreboval využiť. Aby mohla byť aplikácia nahraná na PyPI bolo potrebné pridať niekoľko súborov. Najdôležitejší z nich je „setup.py“, kde sú definované nielen všetky informácie o projekte, ale aj balíčky pre jazyk Python, ktoré sú nevyhnutné pre správny beh aplikácie. V zložkách obsahujúcich skripty v jazyku Python bolo potrebné pridať prázdny súbor „__init__.py“. Ten indikuje, že zložka je súčasťou projektu alebo sa v nej priamo nachádzajú súbory s príponou „.py“. Vďaka tomuto je možné importovať funkcie tohto modulu kľúčovým slovom jazyka Python *import*. Keďže je len malá časť aplikácie Plbmng v jazyku Python a väčšina je napísaná v programovacom jazyku Bash bolo potrebné prekonať pár problémov, ktoré projekty napísané len v jazyku Python nemajú. V súbore „MANIFEST.in“, bolo potrebné definovať všetky súbory, ktoré sú súčasťou projektu a majú byť v ňom zahrnuté pri vytváraní balíčku.



The screenshot shows the PyPI project page for 'plbmng 0.1.8'. The header is blue with a search bar and links for Help, Donate, Log in, and Register. The main section features the project name 'plbmng 0.1.8' in large white text, a green 'Latest version' badge, and a code block with the command 'pip install plbmng'. Below this, it states 'Last released: About 13 hours ago.' and 'Tool for monitoring PlanetLab network'. The page is divided into two columns. The left column contains a 'Navigation' menu with 'Project description' (selected), 'Release history', and 'Download files', followed by 'Project links' with a 'Homepage' link. The right column is titled 'Project description' and contains a 'Description' section. The description states that 'plbmng' is a tool for the PlanetLab network and lists three purposes: getting server information, creating a map, and setting up a cron job for latency/ssh monitoring.

Search projects

Help Donate Log in Register

plbmng 0.1.8 Latest version

`pip install plbmng`

Last released: About 13 hours ago.

Tool for monitoring PlanetLab network

Navigation

- Project description
- Release history
- Download files

Project links

- Homepage

Project description

Description

`plbmng` is a tool created for usage within PlanetLab network. Main purpose of this tool is to manage and monitor your PlanetLab nodes.

For this purpose there are several tools within this project:

- to get all the servers and gather all available information about them
- to create a map with location of the servers
- to set-up a cron job which gets information about latency and/or ssh availability for the selected servers

Obr. 5.2: Stránka aplikácie Plbmng na PyPI.

6 POUŽITIE A ANALÝZA VÝSLEDKOV APLIKÁCIE PLBMNG

Aplikácia pri generovaní výsledkov bola nainštalovaná na virtuálnom stroji s operačným systémom Fedora 28 *Workstation*. Kópia aplikácie Plbmng bola stiahnutá a nainštalovaná z PyPI. Vo výpise 6.1 sú uvedené všetky príkazy, ktoré boli použité v príkazovom riadku pri inštalácii.

```
$ pip3 install plbmng --user

$ sudo dnf install -y dialog pssh fping mc

$ cd
  /home/$USER/.local/lib/python3.6/site-packages/plbmng/

$ ./plbmng.sh
```

Výpis 6.1: Príkazy použité pri inštalácii Plbmng

Prvý príkaz zavola program *pip3*, ktorý podporuje sťahovanie balíkov z PyPI. Ako som spomínal v kapitole 5, pri inštalácii z PyPI stačí jediný príkaz a máme pripravené všetky závislosti, ktoré vyžadujú Python skripty. Nasledujúcim príkazom som si nainštaloval do operačného systému programy:

- Dialog – program na vytváranie okien do shellu zo skriptu v programovacom jazyku Bash [25].
- Midnight Commander – je vizuálny súborový manažér [26].
- PSSH – program umožňujúci nadviazať komunikáciu pomocou protokolu SSH s niekoľkými hosťami naraz [28].
- Fping – umožňuje efektívne odosielať niekoľko ICMP správ naraz [27].

Tretí príkaz vyjadruje presun aktuálneho adresára do zložky plbmng. Toto je východisková cesta pri inštalovaní z PyPI a použitím argumentu pre program *pip3* „user“. Posledný príkaz spustí aplikáciu Plbmng.

Pre fungovanie všetkých funkcií aplikácie je nutné upraviť konfiguračný súbor. Ukážka konfiguračného súboru je vo výpise 6.2. Po zadaní všetkých potrebných údajov je nutné spustiť možnosť *Monitor now* v *Measure menu*. Bez tejto operácie budú využité informácie z predchádzajúceho monitorovania.

```
USERNAME=studentutko@gmail.com
PASSWORD=*****
SLICE=cesnetple_vut_utko
SSH_KEY=/home/$USER/.local/lib/python3.6/site-packages/
        plbmng/bin/private_key
# USERNAME -> Plb web username
# PASSWORD -> Plb web password
# SLICE - > Plb slice name
# SSH -> Full path to your key
# This configuration file can be also modified in the
    settings menu.
```

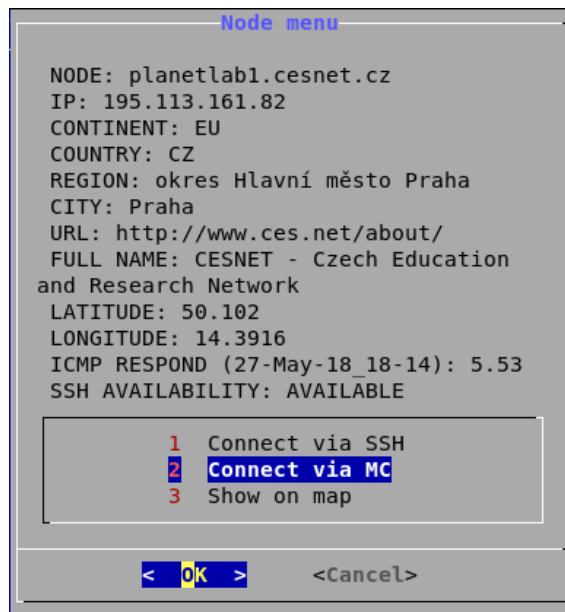
Výpis 6.2: Ukážka vyplneného konfiguračného súboru „plbmng.conf“

Vzdialené pripojenie pomocou aplikácie

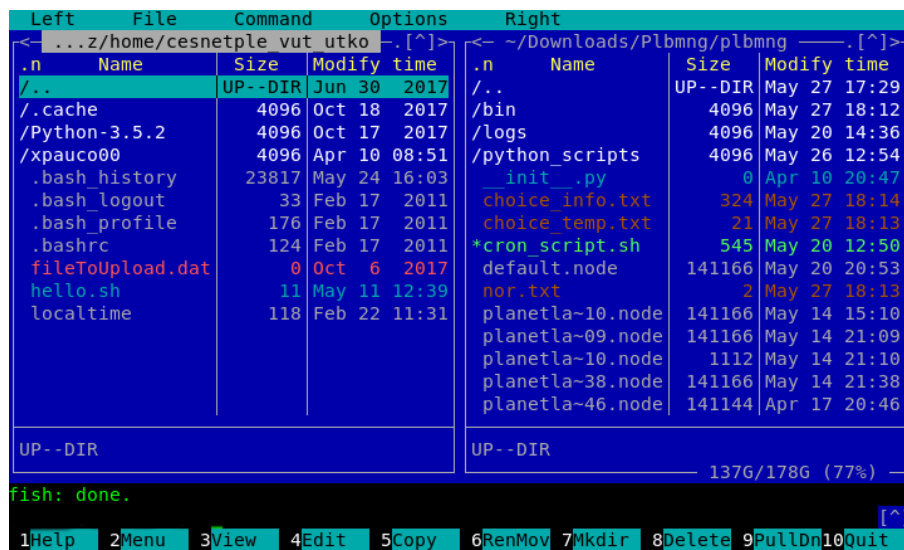
V prípade, že chceme využiť možnosť vzdialeného pripojenia na server pomocou aplikácie, k dispozícii sú dve možnosti. Najprv je potrebné si nájsť požadovaný server pomocou možnosti *Search nodes* v hlavnom menu aplikácie. Menu *Search nodes* v sebe obsahuje 3 možnosti filtrovania všetkých dostupných serverov.

- *Search by DNS* – filtrovanie serverov na základe doménového mena. Zobrazí všetky zhody so zadaným reťazcom.
- *Search by IP* – filtrovanie serverov na základe IP adresy. Zobrazí všetky zhody so zadanou adresou.
- *Search by location* – filtrovanie serverov na základe lokácie. Táto možnosť postupne zobrazí na výber kontinent, krajinu a následne zobrazí všetky dostupné servery z danej krajiny.

Po vybratí serveru sa zobrazí okno so všetkými dostupnými informáciami. Ukážka okna je zobrazená na obrázku 6.1. Dostupné možnosti pripojenia, spomínané na začiatku sekcie, sú pomocou protokolu SSH alebo programu Midnight Commander. Pri generovaní okna je na pozadí prevedený test dostupnosti pripojenia pomocou protokolu SSH. Ak prebehne úspešne v okne sa zobrazí, že je dostupné. Na obrázku 6.2 je ukážka pripojenia na server pomocou programu Midnight Commander.



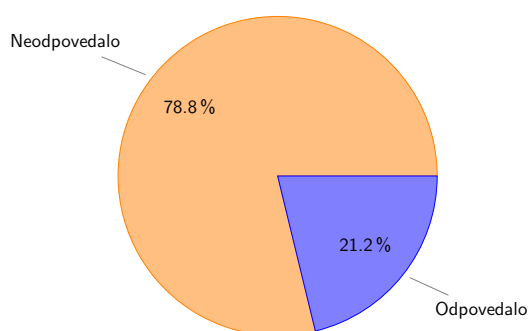
Obr. 6.1: Ukážka okna s informáciami o vybranom serveri.



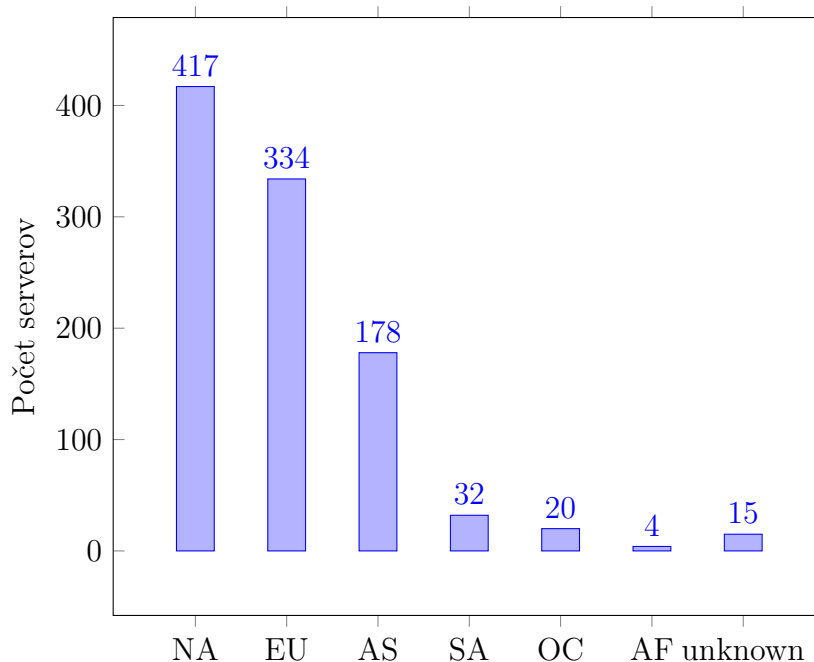
Obr. 6.2: Ukážka programu MC vzdialene pripojeného na server z obrázku 6.1.

Analýza odpovedí serverov na ICMP správu

K tejto analýze bolo dostupných t. č. 1000 serverov. K získaniu údajov bola použitá možnosť *Run measurement now* z *Measure menu*. Graf 6.3 zobrazuje percentuálny pomer serverov, ktoré odpovedali na ICMP správu. Na správu odpovedalo 212 z celkového počtu 1000. Dôvodom, prečo neodpovedalo veľké množstvo serverov, môže byť neschopnosť previesť doménové meno na IP adresu pomocou DNS servera. Tak tiež do úvahy pripadá možnosť zahodenia ICMP paketu firewallom daného serveru z dôvodu, že ICMP správa bola poslaná z neznámej IP adresy. V grafe 6 je zobrazené rozloženie testovaných serverov po svete. Z celkového počtu 1000 serverov nebolo možné určiť kontinent len pri 15, čo je zaujímavý výsledok.



Obr. 6.3: Graf odpovedí serverov na ICMP správu.



Obr. 6.4: Graf umiestnenia serverov na kontinentoch.

7 ZÁVER

Pri písaní práce som sa zoznámil so sieťou PlanetLab jej projektmi a API. Kapitola s operačným systémom Linux a virtualizáciou bola podstatná časť, bez ktorej by nebolo možné rozšíriť aplikáciu a integrovať do nej modul „planetlab_list_creator.py“. Pri vytváraní modulu bolo použité PlanetLab API, bez ktorého by ho nebolo možné zostrojiť. Na získanie lokácie serverov bolo použité reverzné geokódovanie zo služby *OpenStreetMaps*. Modul bol optimalizovaný na efektívne využívanie zdrojov, najmä času. Obyčajné ukladanie údajov získaných o serveroch a neopakovaní žiadostí o reverzné geokódovanie ušetrilo takmer 6 minút času. Modul bol upravený tak, aby ho bolo možné využiť samostatne bez akejkoľvek aplikácie pomocou príkazového riadku. Pri tomto type využitia vráti súbor „default.node“. Aplikácia Plbmng integruje modul pomocou tohto typu využitia, kde mu predáva užívateľské meno a heslo z konfiguračného súboru „plbmng.conf“. Popri práci na integrovaní bola aplikácia rozšírená o funkcie, ktoré mali zjednodušiť a sprehľadniť jej používanie. K aplikácii bol vytvorený manuál, kde je popísaný postup inštalácie a základný prehľad jej funkcií. K modulu a iným skriptom používaným v aplikácii v jazyku Python, bola zostrojená dokumentácia podľa štandardu tohto jazyku a exportovaná ako stránka v jazyku HTML a v čistom texte. Zmeny v aplikácii sú vedené v systéme kontroly verzií na www.github.com. Z tejto stránky je možné získať kópiu spomínanej aplikácie. Aplikácia bola taktiež nahratá na PyPI z dôvodu jednoduchšej možnosti šírenia kópií. Aby bolo možné kópiu nahráť na PyPI, bolo potrebné pridať súbory „MANIFEST.in“ a „setup.py“. Prvý menovaný súbor nie je požadovaný pri nahratí štandardného projektu v jazyku Python na PyPI. Keďže dôležitá časť aplikácie je v jazyku Bash, bolo potrebné deklarovať do tohto súboru názvy súborov, ktoré sú potrebné zachovať pri vytváraní balíku. Súbor „setup.py“ obsahuje informatívne náležitosti o projekte a potrebné balíky v jazyku Python ku správne mu chodu aplikácie. V poslednej časti je popísaný postup pri inštalácii aplikácie a jej využitie pri spravovaní serverov. Vyskytuje sa tam ukážka pripojenia na vzdialený server pomocou programu *Midnight Commander*. Ako posledná sekcia je analýza výsledkov odpovedí serverov na ICMP správu, kde je vynesení do grafu počet serverov na kontinentoch a graf vyjadrujúci pomer serverov, ktoré odpovedali na správu. Tieto údaje boli získané pomocou aplikácie Plbmng a konkrétne funkcie *Run measurement now*, ktorá mala nastavený parameter *Test ping*.

LITERATÚRA

- [1] www.planet-lab.eu. *Planet Lab* [online]. 2010 [cit. 11. 12. 2017].
Dostupné z URL:
<<https://planet-lab.eu/about>>.
- [2] Fiuczynski, M. E.: *User's Guide* [online]. Naposledy aktualizované 02. 01. 2006 [cit. 28. 10. 2017]. Dostupné z URL:
<<https://planet-lab.org/doc/guides/user>>.
- [3] www.gnu.org. *GNU Operating System* [online]. Naposledy aktualizované 16. 11. 2017 [cit. 03. 12. 2017]. Dostupné z URL:
<<https://www.gnu.org/gnu/gnu-history.html>>
- [4] www.planet-lab.org. *PlanetLab Central API* [online]. [cit. 03. 12. 2017].
Dostupné z URL:
<https://www.planet-lab.org/doc/plc_api>
- [5] www.planet-lab.eu. *PlanetLab Central API* [online]. [cit. 03. 12. 2017].
Dostupné z URL:
<<https://www.planet-lab.eu/planetlab/doc/PLCAPI.pdf>>
- [6] ANDRAŠOV, Ivan: *Měření experimentální sítě PlanetLab*. Brno, 2017, 41 s.
Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: doc.Ing. Dan Komosný, Ph.D
- [7] Open Street Map. *Nominatim Usage Policy* [online]. [cit. 03. 12. 2017]. Dostupné z URL:
<<https://operations.osmfoundation.org/policies/nominatim/>>
- [8] Open Street Map. *Nominatim Documentation* [online]. [cit. 10. 05. 2018].
Dostupné z URL:
<https://nominatim.org/release-docs/develop/>>
- [9] Fedora OS. *Fedora Documentation* [online]. [cit. 11. 12. 2017]. Dostupné z URL:
<<https://docs.fedoraproject.org/index.html>>
- [10] CentOS. *CentOS project* [online]. [cit. 11. 12. 2017]. Dostupné z URL:
<<https://www.centos.org/about/>>

- [11] Red Hat, Inc. *Red Hat and the CentOS Project Join Forces to Speed Open Source Innovation* [online]. Naposledy aktualizované 07.01.2014 [cit. 11.12.2017]. Dostupné z URL:
<<https://www.redhat.com/en/about/press-releases/red-hat-and-centos-join-forces>>
- [12] PTACEK, L.: *Linux: dokumentační projekt. 4., aktualiz. vyd.*. Brno: Computer Press, 2007. ISBN 978-80-251-1525-1. [cit. 12.12.2017].
- [13] T. Ylonen: *RFC 4251 - The Secure Shell (SSH) Protocol Architecture*. Technická zpráva, Internet Engineering Task Force, 2006. [cit. 12.12.2017].
- [14] SSH Communications Security, Inc. *SFTP – SSH Secure File Transfer Protocol* [online]. [cit. 12.12.2017]. Dostupné z URL:
<<https://www.ssh.com/ssh/sftp/>>
- [15] www.linux-vserver.org. *Linux VServer* [online]. Naposledy aktualizované 22.04.2013 [cit. 12.12.2017]. Dostupné z URL:
<<http://linux-vserver.org/Overview>>
- [16] CARRIERE, Denis: *Geocoder* [online]. [cit. 10.05.2018]. Dostupné z URL:
<<https://github.com/DenisCarriere/geocoder>>
- [17] [www.github.com](https://github.com). *GitHub* [online]. [cit. 10.05.2018]. Dostupné z URL:
<<https://github.com/about>>
- [18] Open Source Initiative. *MIT license* [online]. [cit. 10.05.2018]. Dostupné z URL:
<<https://opensource.org/licenses/MIT>>
- [19] Zhiyuan Peng, Ranga Reddy Pallela, Haiyang Wang: *On the measurement of P2P file synchronization: Resilio Sync as a case study*. Quality of Service (IWQoS) 2017 IEEE/ACM 25th International Symposium on, s. 1-2 , 2017. [cit. 10.05.2018]. Dostupné z URL:
<<https://ieeexplore.ieee.org/document/5542713/>>
- [20] Guanlin Bian, Yuehui Jin, Tan Yang: *Delay measurement and analysis of network performance on PlanetLab*. Shenzhen, Cloud Computing and Intelligence Systems (CCIS), 2014 IEEE 3rd International Conference, 2017. ISBN 978-1-4799-4720-1. [cit. 10.05.2018].

- [21] I. Norros, V. Pehkonen, H. Reittu, A. Binzenhofer, K. Tutschku: *Relying on randomness PlanetLab experiments with distributed file-sharing protocols*. Next Generation Internet Networks, Trondheim, 2007, s. 6-13. [cit. 10.05.2018]. Dostupné z URL:
<<https://ieeexplore.ieee.org/document/4231814/>>
- [22] E. P. Duarte, T. Garrett, L. C. E. Bona, R. Carmo, A. P. Züge: *Finding stable cliques of PlanetLab nodes*. IEEE/IFIP International Conference on Dependable Systems & Networks (DSN), Chicago, IL, 2010, s. 317-322. [cit. 10.05.2018]. Dostupné z URL:
<<https://ieeexplore.ieee.org/document/5544300/>>
- [23] David Goodger, Guido van Rossum: *Docstring Conventions*, 2001. [cit. 10.05.2018]. Dostupné z URL:
<<https://www.python.org/dev/peps/pep-0257/>>
- [24] www.sphinx-doc.org. *Sphinx Python Documentation Generator* [online]. Dostupné z URL:
<<http://www.sphinx-doc.org/en/master/>>
- [25] Thomas E. Dickey: *Dialog manual* [online]. Dostupné z URL:
<<https://linux.die.net/man/1/dialog>>
- [26] www.midnight-commander.org. *Midnight Commander* [online]. Dostupné z URL:
<<http://midnight-commander.org/>>
- [27] David Schweikert: *Fping* [online]. Dostupné z URL:
<<https://fping.org/>>
- [28] Brent N. Chun, Andrew McNabb: *PSSH manual* [online]. Dostupné z URL:
<<https://linux.die.net/man/1/pssh>>

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

API	Application Programming Interface
TCP	Transmission Control Protocol
ID	Identification
IP	Internet Protocol
URL	Uniform Resource Locator
SSH	Secure Shell
HTML	Hypertext Markup Language
DNS	Domain Name System
GNU	GNU's Not Unix
GPL	General Public License
RHEL	Red Hat Enterprise Linux
SFTP	SSH File Transfer Protocol
CPU	Central Processing Unit
ICMP	Internet Control Message Protocol
MIT	Massachusetts Institute of Technology
PyPI	Python Package Indexer
VServer	Virtual Server
GPG	GNU Privacy Guard
MC	Midnight Commander

ZOZNAM PRÍLOH

A Obsah priloženého CD-R

45

A OBSAH PRILOŽENÉHO CD-R

- Aplikácia Plbmng v archíve typu ZIP
- Dokumentácia
- README
- Modul planetlab_list_creator.py
- Elektronická verzia bakalárskej práce