

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2020

Bc. Filip Šuba



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

SPRÁVA SERVERŮ S OPERAČNÍM SYSTÉMEM FEDORA

MANAGEMENT OF SERVERS WITH FEDORA OPERATING SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Filip Šuba

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. Dan Komosný, Ph.D.

BRNO 2020



Diplomová práce

magisterský navazující studijní obor **Informační bezpečnost**

Ústav telekomunikací

Student: Bc. Filip Šuba

ID: 185944

Ročník: 2

Akademický rok: 2019/20

NÁZEV TÉMATU:

Správa serverů s operačním systémem Fedora

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s aplikací vyvíjenou na Ústavu telekomunikací pro správu serverů s OS Fedora (linuxová distribuce). Aplikace je dostupná na adrese pypi.org/project/plbmng/. Funkce aplikace rozšířte o získávání informací o stavu činnosti severů. Pomocí těchto informací implementujte rozšířené vyhledání serverů podle zvolených kritérií (velikost paměti, dostupný software, verze jádra atd.). Vytvořený kód vystavte pod licencí MIT. Aktualizujte popis aplikace v anglickém jazyce. Aplikaci vystavte na repositáři PyPI.

V rámci semestrálního projektu implementujte vše požadované a proveďte testování s 5 vybranými severy. V rámci diplomové práce proveďte ověření funkčnosti se všemi dostupnými severy.

DOPORUČENÁ LITERATURA:

[1] Linux Dokumentační projekt. 4. vyd. Computer Press, 2008. 1336 s. ISBN: 978-80-251-1525-1.

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

Termín zadání: 3.2.2020

Termín odevzdání: 1.6.2020

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

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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

Diplomová práca sa zaoberá nástrojom pre správu výskumných projektov v celosvetovej experimentálnej sieti PlanetLab. Pre správu je možné využiť aplikáciu Planetlab server manager. V rámci tejto práce bola aplikácia rozšírená o nové funkcie, ktoré predovšetkým sú: získavanie informácií o stave činnosti jednotlivých serverov, rozšírené vyhľadávanie serveru podľa veľkosti operačnej pamäte serveru, verzie jadra OS Linux. Ďalej bola pridaná možnosť správy serverov mimo experimentálnej siete Planetlab a možnosť kopírovania súborov na vybrané servery. Aplikácia je dostupná v repozitári PyPI a zdrojový kód je zverejnený na platforme GitLab pod licenciou MIT. Aplikácia je dostupná pre OS Linux a Mac OS. K aplikácii bol vyhotovený manuál v anglickom jazyku.

KLÚČOVÉ SLOVÁ

Experimentální výskum, Fedora, Linux, PlanetLab, PlanetLab Server Manager

ABSTRACT

The diploma thesis deals with a tool for managing research projects in the global experimental network PlanetLab. It is possible to use the Planetlab server manager application for administration. As part of this work, the application was extended with new functions, which are mainly: obtaining information about the status of operation of individual servers, advanced server search according to the size of the server's operating memory, Linux kernel version. Furthermore, the feature of managing servers outside the Planetlab experimental network and the feature of copying files to selected servers were added. The application is available in the PyPI repository and the source code is published on the GitLab platform under the MIT license. The application is available for Linux and Mac OS. A manual in English was prepared for the application

KEYWORDS

Experimental Research, Fedora, Linux, PlanetLab, PlanetLab Server Manager

ŠUBA, Filip. *Správa serverů s operačním systémem Fedora*. Brno, Rok, 59 s. Diplomová práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: prof. Ing. Dan Komosný, Ph.D.

VYHLÁSENIE

Vyhlasujem, že svoju diplomovú prácu na tému „Správa serverů s operačním systémem Fedora“ som vypracoval samostatne pod vedením vedúceho diplomovej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý 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 zákonníka Českej republiky č. 40/2009 Sb.

Brno

.....

podpis autora

POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce pánovi prof. Ing. Danovi Komosnému, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Obsah

Úvod	11
1 Experimentálna sieť PlanetLab	12
1.1 Terminológia v sieti PlanetLab	12
1.2 Projekty realizované v sieti PlanetLab	13
2 GNU/Linux	15
2.1 Fedora Linux	16
2.2 Využitie SSH na monitorovanie serverov s OS Fedora	16
3 Aktuálny stav aplikácie pre monitorovanie serverov	18
3.1 Pristúpiť k serverom	19
3.2 Monitorovanie serverov	23
3.3 Zobrazíť servery na mape	24
3.4 Set credentials	24
4 Rozšírenie aplikácie	26
4.1 Vyhľadávanie na základe verzií, veľkosti operačnej pamäte	26
4.2 Postup získania informácií zo servera	27
4.3 Využitie tabuľky s informáciami o softvéri a hardvéri	31
4.4 Pridávanie serverov mimo siete PlanetLab	32
4.5 Kopírovanie súborov na servery z databázy aplikácie plbmng	34
4.6 Dokumentácia a refaktorizácia aplikácie plbmng	37
4.7 Zverejnenie aplikácie na platforme GitLab	41
4.8 Zverejnenie aplikácie na repozitári PyPI	44
5 Použitie aplikácie	46
5.1 Aktuálny stav siete PlanetLab	46
5.2 Príklad použitia aplikácie	50
Záver	53
Literatúra	54
Zoznam symbolov, veličín a skratiek	57
Zoznam príloh	58

A	Obsah priloženého CD	59
A.1	Spustenie aplikácie	59
A.2	Zobrazenie dokumentácie	59

Zoznam obrázkov

1.1	Zobrazenie mapy so všetkými servermi experimentálnej siete PlanetLab.	12
2.1	Cesta od zapnutia počítača až po užívateľské prostredie.	15
3.1	Aplikácia plbmng.	18
3.2	Menu „Pristúpiť k serverom“.	19
3.3	Okno s informáciami o serveri.	20
3.4	Pripojenie cez Midnight Commander.	21
3.5	Vygenerovaná mapa zobrazená vo webovom prehliadači Firefox.	22
3.6	Menu „Monitorovanie serverov“.	23
3.7	Ukážka vyplneného konfiguračného súboru v prostredí aplikácie.	25
4.1	Vývojový diagram aktualizácie databázy serverov.	28
4.2	Ukážka vygenerovanej ponuky vyhľadávania na základe verzií jadra OS Linux.	32
4.3	Aktualizované menu aplikácie	33
4.4	Nové menu „Extra“	33
4.5	Ukážka súboru user_servers.node v prostredí aplikácie.	33
4.6	Vývojový diagram vyhľadávania serverov na základne doménového mena.	34
4.7	Výber súboru v prostredí aplikácie.	35
4.8	Výber viac serverov zároveň v prostredí aplikácie.	36
4.9	Zadanie cesty užívateľom pri kopírovaní.	37
4.10	Ukážka úspešne otestovaného kódu z webového rozhrania GitLab.	44
4.11	Ukážka projektovej stránky aplikácie vo webovom rozhraní PyPI	45
5.1	Štatistika serverov experimentálnej siete PlanetLab vytvorená z údajov aplikácie plmbng.	46
5.2	Okno s informáciami o serveri s dostupným portom SSH.	47
5.3	Graf zobrazujúci pomer dostupnosti informácií o serveroch.	48
5.4	Výsledok vyhľadávania na základe regulárneho výrazu 5.4.	51

Zoznam tabuliek

3.1	Výpis riadkov z databázy so statusmi serverov.	24
4.1	Počet stiahnutí aplikácie plbmng za posledných 6 mesiacov.	45
5.1	Verzie jadier zo všetkých dostupných serverov.	49
5.2	Dostupné verzie jazyku Python a veľkosti operačnej pamäte dostupné pre vyhľadávanie v aplikácii.	50

Zoznam výpisov

4.1	Použité príkazy na získanie verzií a veľkosti operačnej pamäte.	27
4.2	Časť kódu funkcie <code>get_server_params</code>	29
4.3	Zobrazenie štandardného výstupu príkazov.	30
4.4	Výpis kódu funkcie <code>run_command</code>	31
4.5	Použitie programu <code>scp</code> získané z manuálu.	34
4.6	Príkaz <code>scp</code> z aplikácie <code>plbmng</code>	35
4.7	Rozdiel pri použití funkcie <code>menu</code> a <code>checklist</code>	36
4.8	Ukážka dokumentácie jednej z funkcií aplikácie <code>plbmng</code>	38
4.9	Ukážka definície modulu určeného na vloženie do dokumentácie. . . .	40
4.10	Skript v jazyku YAML pre účel priebežnej integrácie.	42
4.11	Príkaz na vytvorenie archívu so zdrojovým kódom aplikácie.	44
5.1	Chybová správa pri pokuse o pripojení na server.	47
5.2	Výpis SQL dopytov.	48
5.3	Jednoduchý skript pre účely modelového príkladu využitia aplikácie. .	50
5.4	Regulárny výraz použitý na vyhľadanie konkrétnych serverov.	51
5.5	Výstup skriptu 5.3 spustenom na jednom z vybraných serverov. . . .	52

Úvod

Práca sa zaoberá nástrojmi pre správu výskumných projektov v celosvetovej experimentálnej sieti PlanetLab. Pre komunikáciu s databázou experimentálnej siete PlanetLab je vyvíjaná aplikácia pre monitorovanie serverov s názvom Planetlab server manager (plbmng). Do aplikácie sú pridané funkcie na:

- Získavanie parametrov serverov (veľkosť operačnej pamäte, verzia jadra Linux, verzia programovacieho jazyka Python a verzia kompilátoru programovacieho jazyka C).
- Vyhľadávanie serverov na základe parametrov získaných z predchádzajúceho bodu.
- Možnosť vkladať do databázy servery mimo siete PlanetLab.
- Podpora kopírovania na servery z databázy aplikácie plbmng.

Ako minoritné vylepšenie aplikácie je vylepšená jej čitateľnosť. Tá je zlepšená rozdelením jediného súboru s kódom na tri ďalšie podľa logických celkov. Ďalej bola vykonaná úprava kódu na základe štandardu PEP-0008, ktorý určuje detaily vylepšujúce práve čitateľnosť. Všetky rozšírenia a vylepšenia aplikácie sa nachádzajú podrobne popísané v kapitole 4.

Modifikovaný zdrojový kód je následne zverejnený na platforme GitLab s podporou priebežnej integrácie. Pre podporu priebežnej integrácie je vytvorený skript v jazyku YAML s definovanými úlohami, ktoré slúžia ako napr. testy. Úlohy použité v práci slúžia na testovanie a zverejnenie dokumentácie na webovej stránke. Hotový projekt je zverejnený na repozitári PyPI pod licenciou MIT. Popis aplikácie vrátane manuálu v anglickom jazyku je aktualizovaný na základe pridaných rozšírení a zverejnený spolu so zdrojovým kódom.

V poslednej kapitole 5 je vykonané meranie v rámci celej siete PlanetLab a vyhodnotené výsledky z aplikácie plbmng. Vyhodnocujú sa v nej výsledky o dostupnosti serverov a ich parametroch, ktoré boli získané pridanými rozšíreniami v kapitole 4. Na konci práce v sekcii 5.2 je opísané použitie aplikácie na modelovom príklade.

1 Experimentálna sieť PlanetLab

PlanetLab je verejne dostupná celosvetová experimentálna sieť, ktorá poskytuje podporu pri vývoji nových internetových služieb a aplikácií. Konzorcium PlanetLab sa sformovalo v roku 2002, ktorého členmi boli akademické inštitúcie, firmy pracujúce v odbore a vládne organizácie. Za dohľad a rast experimentálnej siete je zodpovedné toto konzorcium [1]. V decembri 2011 pozostávala experimentálna sieť PlanetLab z 1024 uzlov a 530 rôznych inštitúcií po celom svete [2]. Na obrázku 1.1 sú zobrazené všetky uzly tejto siete. Obrázok 1.1 bol vytvorený aplikáciou plbmng, ktorá je popísaná v kapitole 3.



Obr. 1.1: Zobrazenie mapy so všetkými servermi experimentálnej siete PlanetLab.

1.1 Terminológia v sieti PlanetLab

Pre pochopenie tejto práce a porozumeniu experimentálnej siete PlanetLab je potrebné vysvetliť nasledujúce pojmy. Najčastejšie používané pojmy boli vybrané z dokumentácie siete PlanetLab [3]:

- Miesto – Miesto, kde sa nachádzajú fyzicky uzly experimentálnej siete PlanetLab. Pod miestom si je možné predstaviť akademické inštitúcie, firmy prípadne vládne organizácie.
- Uzol – Uzol, inak povedané server. Server, patriaci do experimentálnej siete PlanetLab, sa odlišuje oproti ostatným bežným serverom len tým, že riadi všetky vyžadované komponenty siete PlanetLab.

- Časť – Je to množina, v ktorej sa nachádzajú všetky uzly, ktoré si užívateľ do svojej časti pridá. Táto časť musí mať svoj jedinečný názov. Pri pripojení sa na akýkoľvek z uzlov užívateľom definovanej časti, sa prihlasuje práve týmto názvom. Pri pridaní uzlu do časti je automaticky synchronizovaný názov časti a verejný kľúč, ktorý patrí k tejto časti.
- Fragment – Je to množina prostriedkov alokovaná na jednom uzle v experimentálnej sieti PlanetLab.
- Virtuálny server – Fragmenty sú v rámci siete alokované pomocou Linux-Vservers. Je to jedno z riešení, ktoré je dnes možné použiť pre vytváranie a správu virtuálnych serverov.

1.2 Projekty realizované v sieti PlanetLab

Od vzniku experimentálnej siete bola využitá na množstvo projektov a pomohla vyvinúť mnoho užitočných aplikácií. Podľa projektov je ľahšie určiť, na čo sa konkrétne využíva. Avšak nie všetky realizované projekty využívajú sieť, len pre svoje vylepšenie. Preto budú v nasledujúcej časti spomenuté a v krátkosti popísané niektoré vybrané projekty uskutočnené v sieti PlanetLab. Ako bude spomenuté, niektoré z projektov využili sieť a zároveň sa snažili dosiahnuť vylepšenie určitých funkcií v nej.

Každodenné meranie oneskorenia a analýza výkonu siete vykonaná v sieti PlanetLab

Práca sa zaoberá výkonom siete a hovorí, ako je dôležité merať tento parameter pre lepšie využívanie zdrojov v budúcnosti. Výkon siete môže byť odlišný v rôznych prostrediach, čo značne narušuje kvalitu aplikácií, ktoré vyžadujú kvalitnú sieťové spojenie. Na základe distribuovaného merania výkonu siete je možné vďaka celosvetovej sieti PlanetLab získať charakteristiku oneskorenia naprieč svetom pomocou časovej a priestorovej analýzy [4].

Experimentálne vyhodnotenie súkromie zachovávajúcich agregáčnych schém v sieti PlanetLab

Práca poukazuje na skutočnosť, že mnohé súčasné technológie odhaľujú užívateľské návyky, čo narušuje súkromie a bezpečnosť ľudí. Autori tejto práce preto navrhli riešenie pomocou myšlienky súkromie zachováajúceho ťaženia dát, ktoré obsahujú dáta o niekoľko užívateľoch naraz, bez možnosti zistenia dát jednotlivca. Protokoly tohto typu ale môžu trpieť na výpadok niektorých dát počas procesu ich

operácie. Preto experimentálna sieť PlanetLab poslúžila ako testovacie prostredie pre aplikovanie protokol tohoto typu [5].

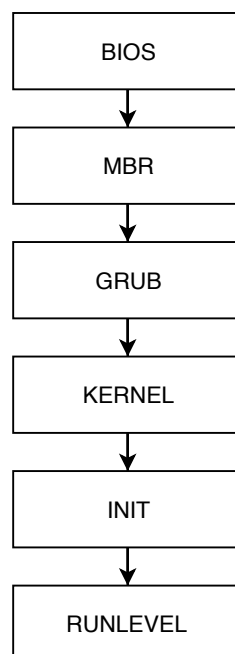
Migrácia virtuálnej siete na reálne infraštruktúre: PlanetLab prípadová štúdia

Práca pojednáva o virtualizácii siete, ktorá vo svojej podstate umožňuje na súčasnej infraštruktúre prevádzku nových sieťových architektúr a aplikácií. Pre zabezpečenie efektívneho využívania zdrojov, musia niekedy virtuálne siete meniť ich pozíciu v sieti. V rámci práce vytvorili autori nástroj, ktorý riadi presúvanie a rozmiestňovanie virtuálnych sietí na nové pozície. Na konci práca vydáva doporučenia o všeobecnej virtualizácii prostredia a špecifické rady pre experimentálnu sieť Planetlab, ktoré umožnia práve presúvanie virtuálnych sietí [6].

2 GNU/Linux

GNU/Linux je operačný systém založený na jadre s názvom Linux, ktoré vytvoril Linus Benedict Torvalds počas svojho štúdia na Univerzite v Helsinkách. Prvýkrát sa o jadre zmienil v roku 1991. Jadro nakoniec daroval svetu, pretože ho vydal ako projekt s otvoreným zdrojovým kódom pod licenciou GPL. Projekt pod licenciou GPL môže ktokoľvek na svete modifikovať, zdieľať, spúšťať a dokonca modifikovaný softvér je po splnení určitých podmienok možné aj predávať. To, čo sa na začiatku tváril ako malý voľnočasový projekt, je dnes najpoužívanejším jadrom operačného systému na serveroch [7].

Vela ľudí chybne označuje názvom Linux celý operačný systém. Linux je len jadro, ktoré ovláda hardvérové zdroje a umožňuje aplikáciám ich využívať. Veľkú časť z operačného systému tvoria programy projektu GNU. Projektu GNU chýbalo jadro, ktoré neskôr pridal do projektu Linus Torvalds. Po pridaní Linusovho jadra bol konečne systém GNU kompletný, aj keď stále bolo potrebné upraviť množstvo ostatných programov, ktoré boli súčasťou projektu GNU pred pridaním Linuxu. Aj keď tento proces nebol jednoduchý, dnes je celý svet vďačný všetkým programátorom, ktorí sa podieľali na vytvorení bezplatného operačného systému [7]. Pre lepšie pochopenie, kde sa nachádza jadro v tomto projekte, je na obrázku 2.1 zobrazený proces od zapnutia počítača až po nabehnutie všetkých častí operačného systému [8].



Obr. 2.1: Cesta od zapnutia počítača až po užívateľské prostredie.

2.1 Fedora Linux

Ako bolo spomenuté v predchádzajúcej sekcii, GNU/Linux má otvorený zdrojový kód a je vydaný pod licenciou GPL, čo umožňuje firmám alebo osobám vytvoriť svoju vlastnú verziu operačného systému GNU/Linux. Vlastné verzie firiem prípadne osôb sa nazývajú slovom distribúcie. Distribúcie sa odlišujú podporovanou platformou procesoru, užívateľským rozhraním atď.

Distribúcia, ktorá sa používa v rámci siete PlanetLab, sa nazýva Fedora. Fedora Linux je distribúcia bežiacia na jadre Linux s otvoreným zdrojovým kódom. Je vytváraný komunitou pre komunitu. Fedora funguje ako platforma, kde firma Red Hat poskytuje najnovšie inovácie, ktoré sa možno raz dostanú do Red Hat Enterprise Linux.

Red Hat Enterprise Linux je podobne ako Fedora distribúcia s otvoreným zdrojovým kódom, ale nie je tak úplne zadarmo. Ak chce používateľ v Red Hat Enterprise Linux využívať všetky jeho výhody, potrebuje si platiť predplatné. Red Hat Enterprise Linux sa zameriava hlavne na stabilitu a podporu pre zákazníkov, ktorú poskytuje na 10 rokov. Týmto je zaručené, že užívateľ má stabilný systém, ktorý dostáva len bezpečnostné záplaty a iné vylepšenia pri životnom cykle softvéru. Táto veta preto umožňuje lepšie pochopiť prečo Red Hat Enterprise Linux potrebuje Fedoru.

Aby bol softvér stabilný, potrebuje používateľov, ktorí ho používajú. Je málo pravdepodobné, že používateľ, ktorý chce mať stabilný systém, bude experimentovať s novými funkciami, pri ktorých hrozí pád systému. Preto je Fedora sponzorovaná spoločnosťou Red Hat. Toto sponzorstvo prináša úžitok pre komunitu tým, že inžinieri z firmy Red Hat prispievajú do Fedory najnovšie inovácie. A podobne pre firmu Red Hat, ktorá prináša svojim platiacim zákazníkom operačný systém, ktorý je viac stabilný na základe odstránenia bežných chýb pri používaní vo Fedore, s ktorými vývojár inovácie nepočítal. To pravdaže neznamená, že nové inovácie sú nestabilné, a tým pádom aj celá distribúcia Fedora Linux. Na základe týchto informácií je preto jasné, že Fedora je správnou voľbou pre experimentálnu sieť PlanetLab, pretože výskumník má k dispozícii všetky najnovšie inovácie [9], [10].

2.2 Využitie SSH na monitorovanie serverov s OS Fedora

Protokol SSH je dnes neodmysliteľnou súčasťou každodennej práce systémového administrátora. V distribúcii Fedora sa nachádza balíček **OpenSSH**, v ktorom sa nachádza implementácia protokolu SSH s otvoreným zdrojovým kódom. Obsahuje

množinu nástrojov, ktoré umožňujú vzdialené pripojenie pomocou protokolu SSH. Šifruje celú komunikáciu medzi bodom A a B, aby boli znemožnené útoky typu MITM, nabúranie spojenia a tak ďalej. Ako príklad funkcií nachádzajúcich sa v **OpenSSH** boli vybrané nasledovné [11]:

- Šifrovaná komunikácia, ktorá začína pred autentizáciou a žiadne informácie vrátane hesiel nie sú posielané ako čistý text. Používajú sa na to algoritmy ako napr. AES, RSA, ECDSA.
- Poskytovanie autentizácie, ktorá má chrániť proti rôznym druhom útokov, kde sa útočník snaží tváriť ako niekto iný a získať z toho prospech. Ako ochrana proti tomuto druhu útokov sa používajú verejné kľúče, prípadne heslá na jedno použitie.
- Poskytovanie možnosti predávania portov na vzdialený cieľ cez zašifrovaný kanál, čo umožňuje aj pri protokoloch bez šifrovania, teda používať šifrovanie.

Vybraná ukážková množina nástrojov, ktoré sú súčasťou **OpenSSH** [12]:

- ssh – Klientská časť nástroja na vzdialené pripojenie.
- sshd – Serverová časť nástroja na vzdialené pripojenie tzv. daemon.
- scp – Nástroj na bezpečné vzdialené kopírovanie.
- ssh-add – Umožňuje pridať privátny kľúč užívateľa na vzdialeného hostiteľa.
- ssh-keygen – Nástroj na generovanie autentizačných kľúčov.

Na monitorovanie serverov sa hlavne používa klientská a serverová časť nástroja **OpenSSH**. Klient posiela zabezpečeným kanálom príkaz na server. Spolu s príkazom preukazuje svoju totožnosť privátnym kľúčom. Ak sa nachádza na serveri medzi autorizovanými kľúčmi verejný kľúč klienta, je mu umožnený príkaz vykonať. Príkaz sa vykoná na serveri, ale klient prijíma výstup a môže s ním ďalej pracovať. Toto je využívané v kapitole 4 pri zbere dát o softvéri/hardvéri na serveri [13].

3 Aktuálny stav aplikácie pre monitorovanie serverov

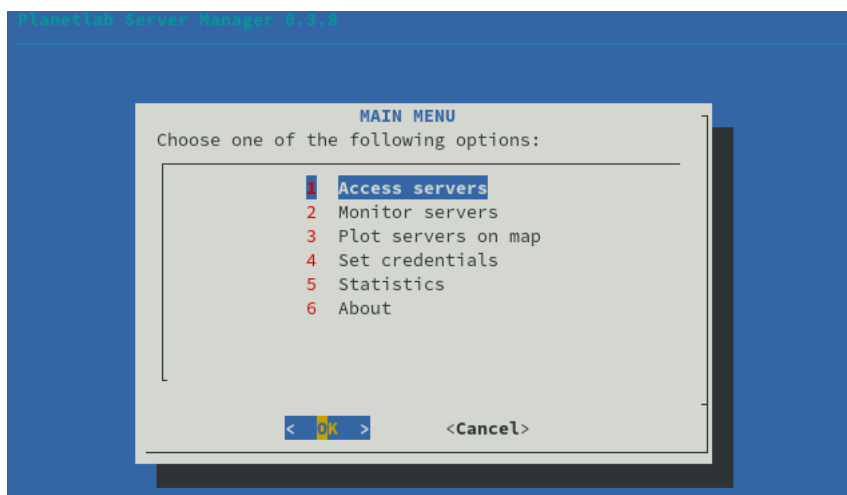
Aplikácia plbmng je nástroj na monitorovanie a správu serverov v rámci siete experimentálnej siete PlanetLab. Posledná dostupná verzia z PyPI je „0.3.8“ a hlavný výstup tejto diplomovej práce bude postavený na tejto verzii. Aplikácia je kompletne napísaná v jazyku Python, ako rozhranie je použité textové užívateľské rozhranie s názvom **Dialog**.

Prvá verzia bola vytvorená v rámci bakalárskej práce Ivana Andrašova [14]. Rozšírenie aplikácie od Ivana Andrašova prebehlo v rámci mojej bakalárskej práce [15]. Posledná verzia tejto aplikácie bola vytvorená v rámci diplomovej práce Martina Kačmarčíka [16]. Aplikácia bola vytvorená za účelom zjednodušenia prístupu k zdrojom experimentálnej siete PlanetLab.

Popis funkcionality aplikácie

Po spustení aplikácie je užívateľ vyzvaný na vyplnenie prihlasovacieho mena a hesla do siete PlanetLab, cesty k súboru obsahujúceho privátny kľúč a názov užívateľa, pod ktorým sa prihlasuje na jednotlivé uzly (servery) experimentálnej siete PlanetLab.

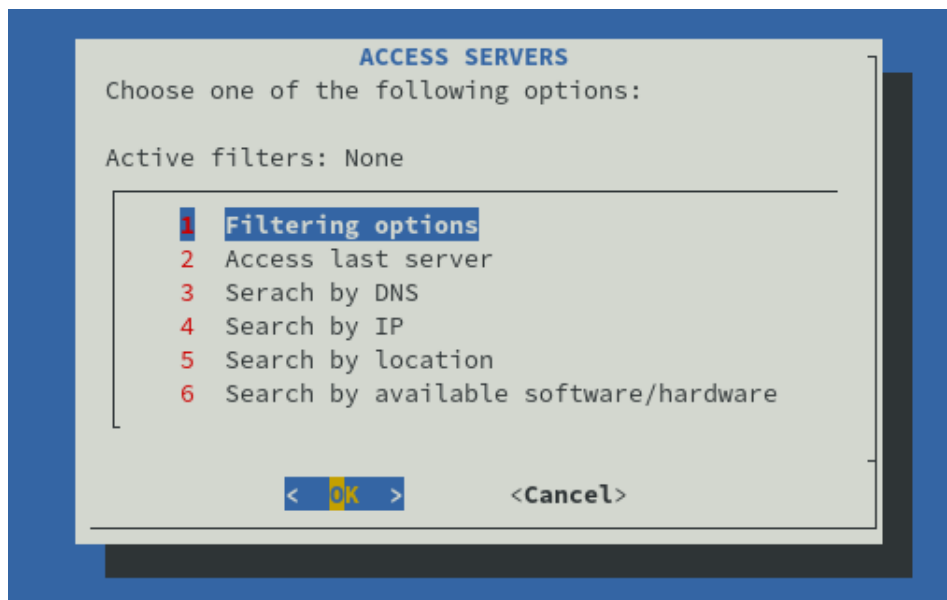
Všetky tieto údaje sú potrebné pre využitie všetkých dostupných funkcií tejto aplikácie. Tieto údaje si vytvára užívateľ pri registrácii do tejto siete a má možnosť kedykoľvek si ich zmeniť v nastaveniach na webovej stránke experimentálnej siete PlanetLab. Základné menu aplikácie je možné vidieť na obrázku 3.1.



Obr. 3.1: Aplikácia plbmng.

3.1 Pristúpiť k serverom

Prvá možnosť v základnom menu slúži na prístup k jednotlivým serverom. Po kliknutí na túto možnosť sa užívateľovi zobrazí ďalšie menu, v ktorom sú pripravené možnosti vyhľadávania. Umožnenie vyhľadávania serverov je jedna z najdôležitejších funkcií tejto aplikácie. Ukážka možností vyhľadávania je na obrázku 3.2

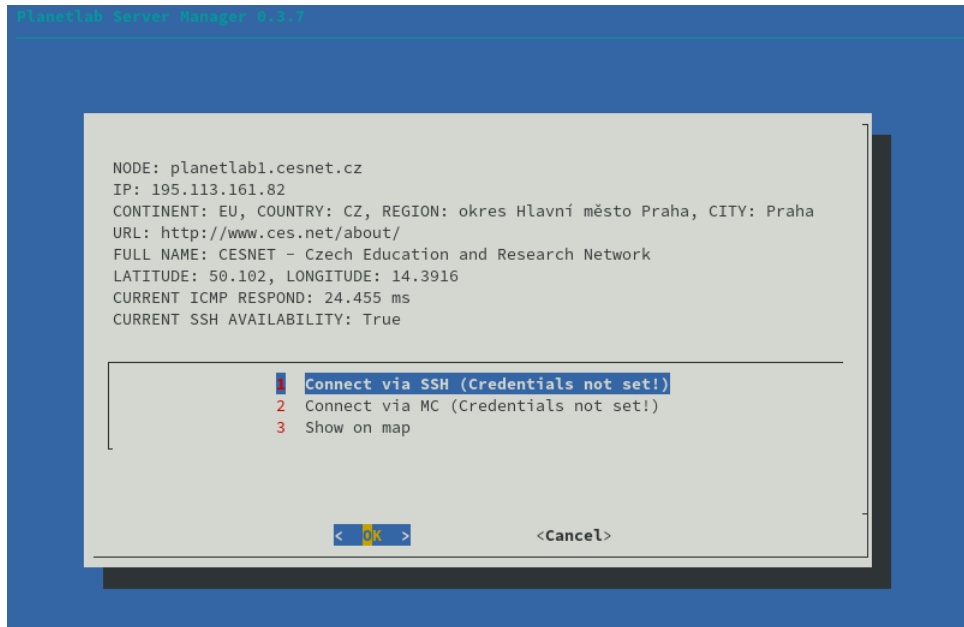


Obr. 3.2: Menu „Pristúpiť k serverom“.

Popis jednotlivých možností filtrovania:

1. Posledný prístupný server – Po zvolení tejto možnosti sa zobrazí užívateľovi okno s informáciami o poslednom sprístupnenom serveri. Ukážka tohoto typu okna je na obrázku 3.3.
2. Vyhľadávať podľa DNS – Pri použití tejto možnosti užívateľ zadá do nasledujúceho okna reťazec znakov. V prípade zhody, sa v ďalšom okne zobrazia všetky servery, v ktorých doménovom mene sa nachádza zadaný reťazec znakov.
3. Vyhľadávať podľa IP – Tak ako pri predchádzajúcej možnosti, aj tu je užívateľ vyzvaný k zadaniu znakov. V prípade nájdenia zhody, sa mu zobrazia všetky IP adresy, ktoré sa zhodujú so zadaným reťazcom znakov.
4. Vyhľadávať podľa polohy – Užívateľ si postupne vyberá z možností, ktoré sú pripravené vopred. Najprv vyberá kontinent, potom krajinu a na základe týchto dvoch odpovedí sú zobrazené všetky doménové mená serverov v krajine, ktorú si vybral.

5. Možnosti vyhľadávania – V tejto možnosti si užívateľ môže obmedziť množinu serverov na tie, ktoré odpovedali na PING alebo majú dostupný port s číslom 22, alebo kombináciu predchádzajúcich možností. Dostupnosť portu 22 sa kontroluje, pretože ak je port nedostupný, je nemožné otvoriť spojenie pomocou protokolu SSH.



Obr. 3.3: Okno s informáciami o serveri.

Výsledok celého tohoto filtrovacieho procesu je vybratie konkrétneho serveru a zobrazenie informácií o ňom. Na obrázku 3.3 je zobrazený výsledok úspešného filtrovacieho procesu. Toto okno obsahuje nasledujúce informácie:

- Node – Doménové meno
- IP – IP adresa serveru
- Continent – Kontinent
- Country – Krajina, kde sa server nachádza
- Region – Región
- City – Mesto
- URL – Odkaz na webovú stránku poskytovateľa serveru do experimentálnej siete PlanetLab
- Full name – Celé meno poskytovateľa
- Latitude – Zemepisná šírka polohy serveru
- Longitude – Zemepisná dĺžka polohy serveru
- Current ICMP Respond – Odozva na ping
- Current SSH availability – Dostupnosť pripojenia pomocou SSH

V prípade, že nejaký z týchto parametrov nie je možné zistiť, je nahradený slovom „unknown“.

Všetky vyššie spomenuté informácie okrem „Current ICMP Respond“ a „Current SSH availability“ sa čerpajú zo súboru „default.node“. Tento súbor je výsledkom skriptu s názvom planetlab_list_creator.py, ktorý bol pridaný do tejto aplikácie v rámci mojej bakalárskej práce [15].

Ďalej sa v okne nachádzajú tri možnosti, ktoré sú podstatou tejto aplikácie:

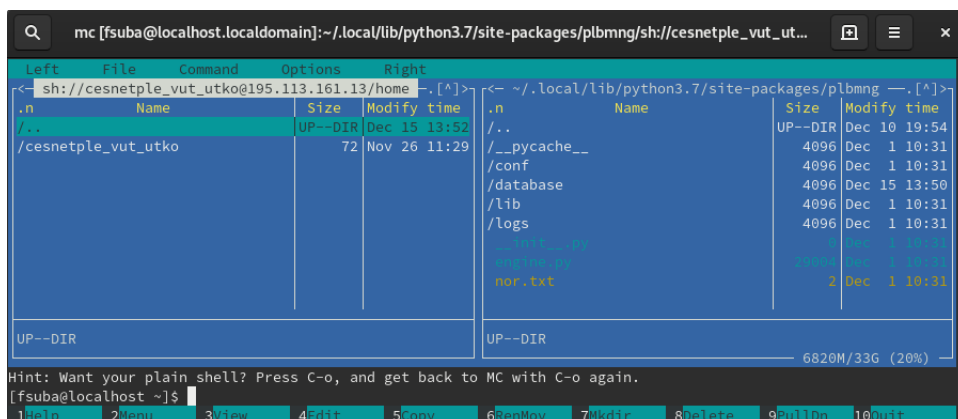
1. Pripojiť cez SSH
2. Pripojiť cez MC
3. Zobrazíť na mape

Pripojiť cez SSH

Táto možnosť sa pokúsi pripojiť na konkrétny server pomocou protokolu SSH. Ak je pripojenie úspešné, automaticky sa otvorí terminál a okamžite je možné pracovať so vzdialeným serverom. Po ukončení užívateľom je automaticky vrátený do začiatočného menu pre vyhľadávanie 3.2.

Pripojiť cez MC

Midnight commander, skrátene MC, je vizuálny prehliadač súborov pre terminál¹. Tento nástroj je často používaný medzi administrátormi pre ľahšiu orientáciu v súborovom systéme. Z tohoto dôvodu bol pridaný aj do aplikácie plbmng. Na obrázku 3.4 je zobrazená ukážka pripojenia na jeden z dostupných serverov siete PlanetLab. Tak ako pri predchádzajúcej možnosti, užívateľ po ukončení pripojenia je vrátený do začiatočného menu pre vyhľadávanie 3.2.



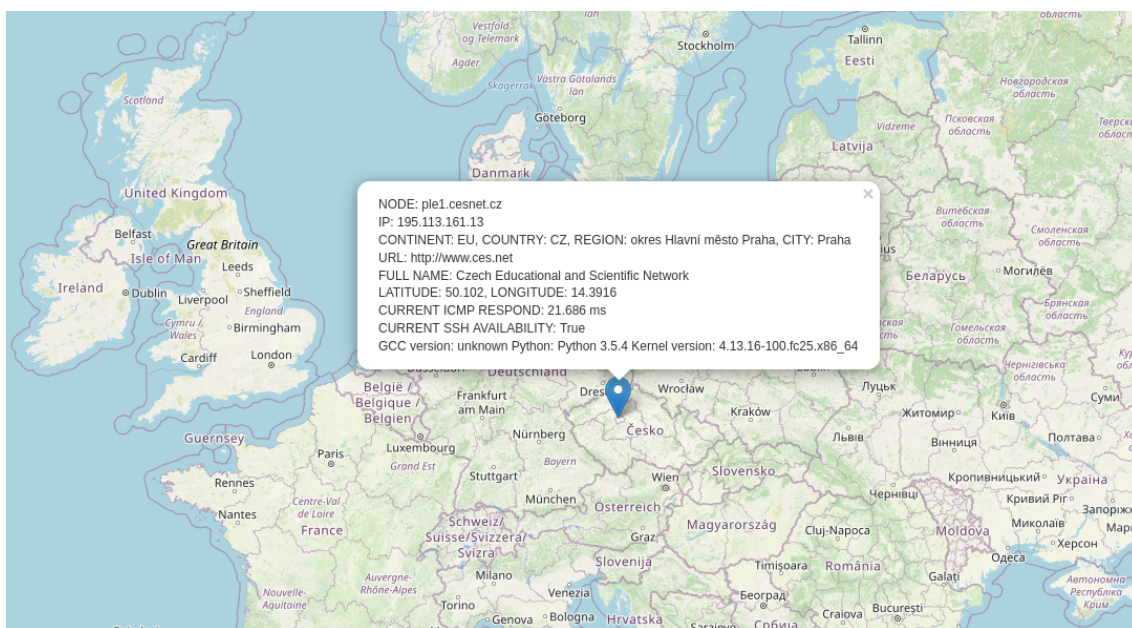
Obr. 3.4: Pripojenie cez Midnight Commander.

¹<https://midnight-commander.org/>

Zobraziť na mape

Na základe informácií o serveri, a to konkrétne zemepisnej šírke a zemepisnej dĺžke, je na mapu vynesovaný bod, ktorý znázorňuje približnú polohu serveru. Táto mapa je vygenerovaná ako webová stránka v jazyku HTML, uložená na disku a následne otvorená základným webovým prehliadačom. Ukážka tejto mapy zobrazenej v prehliadači Firefox je na obrázku 3.5.

Tieto možnosti sú k dispozícii vždy, aj keď je server nedostupný, prípadne nie sú zadané prihlasovacie údaje ako na obrázku 3.3. Chýbajúce prihlasovacie údaje sú signalizované správou v zátvorke. Ako indikátor nedostupnosti serveru, prípadne len nedostupnosti pripojenia pomocou protokolu SSH, nám slúžia informácie „Current ICMP Respond“ a „Current SSH availability“. Ak sa za informáciou „Current ICMP Respond“ nenachádza odozva v milisekundách, znamená to, že server je nedostupný. V tom prípade sa aj za „Current SSH availability“ objaví booleanovská hodnota „False“.

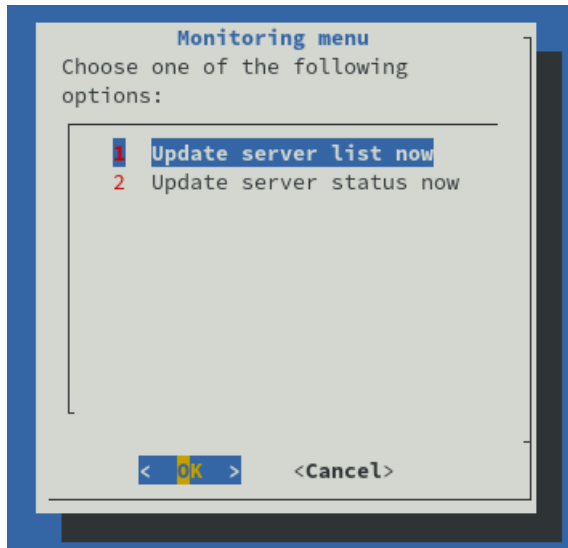


Obr. 3.5: Vygenerovaná mapa zobrazená vo webovom prehliadači Firefox.

3.2 Monitorovanie serverov

Táto možnosť v hlavnom menu slúži na aktualizáciu databázy serverov a aktualizáciu statusu jednotlivých serverov. Ako je možné vidieť na obrázku 3.6, nachádzajú sa tu tri možnosti:

1. Okamžitá aktualizácia zoznamu serverov – Okamžite spustí aktualizáciu databázy serverov. Po vybratí možnosti sa užívateľovi zobrazí varovanie, že tento proces môže trvať až 20 minút. V tomto čase je terminál zablokovaný a užívateľ nemôže ďalej využívať toto okno. Ak chce užívateľ naďalej používať program, aj keď ešte s nie aktuálnou databázou, je možné si otvoriť ďalší terminál, a tam spustiť plbmng znovu.
2. Okamžitá aktualizácia serverového statusu – Táto možnosť spustí aktualizáciu databázy, kde sú uložené informácie o odpovediach jednotlivých serverov na ICMP žiadosť a či je dostupné pripojenie pomocou protokolu SSH. Ukážka tejto databázy sa nachádza v tabuľke 3.1. V nej sú vždy k doménovému menu serveru priradené dve hodnoty, „T“ ako „True“ a „F“ ako „False“. Prvý stĺpec po doménovom mene je výsledok na žiadosť ICMP a druhý stĺpec na dostupnosť pripojenia pomocou protokolu SSH. Tento proces prebieha súbežne pre niekoľko serverov. Taktiež ako v predchádzajúcej možnosti je terminál zablokovaný a je nutné preto počkať, kým nie je tento proces dokončený.



Obr. 3.6: Menu „Monitorovanie serverov“.

Tab. 3.1: Výpis riadkov databázy so statusmi serverov.

nkey	shash	shostname	bssh	bping
10	b1b403fc908b82a4ed7a75ff3e999da8	planetlab2.sics.se	F	F
11	ff37f9b8b4506701fad4043809c5c3e4	vicky.planetlab.ntua.gr	T	T
12	3f3c0682fb1d6cb0da277a2d171d1924	planetlab1.virtues.fi	F	T
13	d0519bb9ff8f167ceb6c0b39cfb5ab5a	planetlab1.sics.se	F	F

3.3 Zobrazit servery na mape

Ako tretia možnosť v aplikácii plbmng sa nachádza vykreslenie mapy so servermi. Po otvorení tejto možnosti sú používateľovi poskytnuté tri možnosti filtrovania serverov, ktoré sa budú na mape zobrazovať. Tieto možnosti sú nasledovné:

- Plot servers responding to ping – Táto možnosť vykreslí na mapu len servery, ktoré reagovali na ICMP žiadosť.
- Plot ssh available servers – Vykreslenie všetkých serverov na mapu, pri ktorých je dostupné pripojenie pomocou protokolu SSH.
- Plot all servers – Vykreslenie všetkých serverov v databáze.

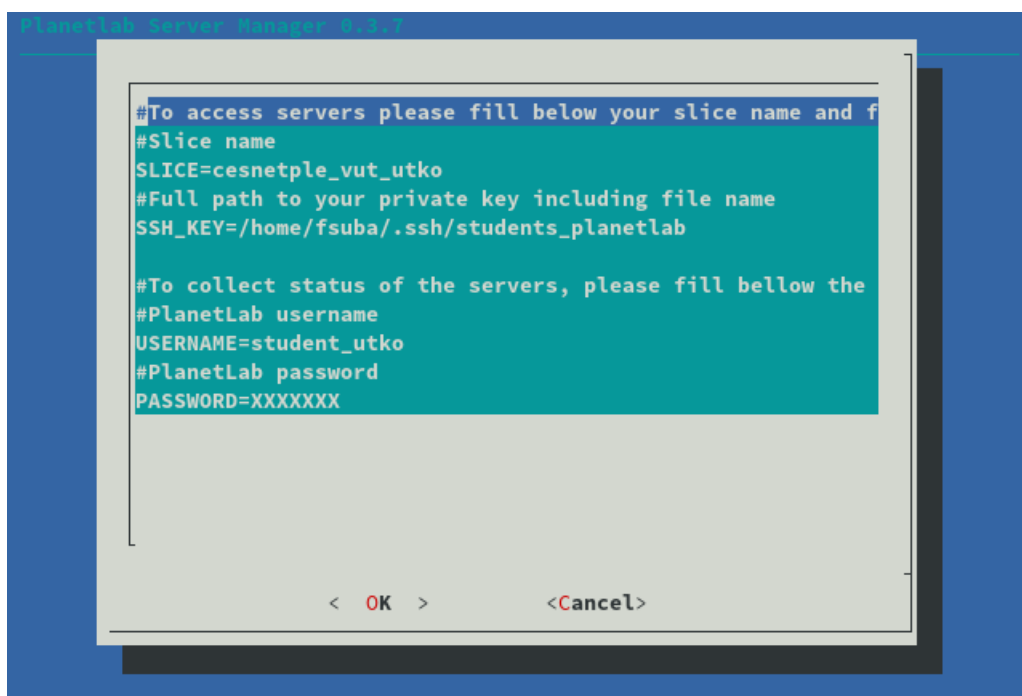
Všetky z týchto možností vygenerujú mapu, ktorú je možné otvoriť pomocou webového prehliadača. Na obrázku 1.1 je ukážka vygenerovanej mapy so všetkými servermi v databáze pomocou možnosti „Plot all servers“.

3.4 Set credentials

Túto možnosť môžeme vyhodnotiť ako najdôležitejšiu zo všetkých. Nesie v sebe všetky dôležité údaje, ktoré vyplňa používateľ, aby mohol pristupovať k serverom. Po otvorení menu sa zobrazí textový editor, ktorý otvára konfiguračný súbor. Tento súbor je možné upravovať aj bez toho, aby bolo potrebné aplikáciu plbmng spúšťať. Pravdaže pre jednoduchšie použitie je možné modifikovať tento súbor priamo v aplikácii. Súbor obsahuje nasledujúce štyri parametre, ktoré je možné vidieť aj na obrázku 3.7:

- „SLICE“ – Pod týmto názvom má užívateľ definovanú množinu serverov, na ktorých sa mu vytvorí lokálny účet. Pri pripojení sa na ktorýkoľvek z týchto serverov, sa používa ako prihlasovacie meno.

- „SSH_KEY“ – Užívateľ si vyplní cestu k súboru, v ktorom sa nachádza privátny kľúč, na základe ktorého mu je umožnené prihlásenie bez nutnosti zadania hesla. Verejný kľúč je potrebné zadať v nastavení profilu na webovom rozhraní siete PlanetLab a následne je z nej synchronizovaný na všetky servery v ňom špecifikovanej množine.
- „USERNAME“ – Prihlasovacie meno na webové rozhranie siete Planetlab. Tento údaj je používaný pri aktualizácii databázy serverov a je nevyhnutný, aby táto operácia prebehla úspešne.
- „PASSWORD“ – Heslo, ktoré patrí k prihlasovaciemu menu. Tak ako pri „USERNAME“, aj toto je nevyhnutný parameter pre aktualizáciu databázy serverov.



Obr. 3.7: Ukážka vyplneného konfiguračného súboru v prostredí aplikácie.

4 Rozšírenie aplikácie

Pri používaní aplikácie užívatelia narážajú na limity aplikácie. Užívatelia poskytujú spätnú väzbu na základe ktorej, je možné vytvoriť prostredie vyhovujúce ich potrebám. Týmto spôsobom boli identifikované rozšírenia:

- Vyhľadávanie v databáze na základe vopred vybraných parametrov
- Pridávanie serverov mimo siete PlanetLab do databázy aplikácie plbmng
- Možnosť kopírovania súboru/ov na užívateľom vybraný server/y

Celá táto kapitola sa opiera o kapitolu 3, kde bola opísaná a znázornená kompletná funkcionálna aplikácia plbmng. Hlavným dôvodom pre pridanie rozšírenia boli problémy pri výbere serverov, ktoré by boli schopné bežať určité aplikácie. Ako konkrétny príklad môžem spomenúť problémy pri inštalácii NS3. Program NS3 vyžaduje minimálne požiadavky, ktoré niektoré servery v rámci experimentálnej siete PlanetLab nedosahujú. Preto bolo navrhnuté riešenie, ktoré ušetrí užívateľovi čas stratený náhodným výberom typu pokus, omyl. Riešenie pridané do aplikácie zbiera informácie o vopred vybraných parametroch serverov v databáze aplikácie plbmng.

4.1 Vyhľadávanie na základe verzií, veľkosti operačnej pamäte

Táto funkcionálna je pridaná do menu opísanom v sekcii 3.1 s názvom *Pristúpiť k serverom*. Na ukladanie údajov sa použije databáza sqlite, ktorá sa už nachádza v aplikácii a úryvok z nej sa nachádza na obrázku 3.1. Pre ľahšiu údržbu a čitateľnosť je kladený dôraz na zachovanie formátu, ktorý je použitý pri uchovávaní informácií o stave serverov. Vyhľadávanie serverov bude prebiehať na základe verzií a veľkosti operačnej pamäte.

Pre vyhľadávanie sú použité nasledovné položky:

- Kompilátor programovacieho jazyka C na platforme Linux GCC.
- Programovací jazyk Python.
- Jadro operačného systému Linux.
- Operačná pamäť serveru.

Na základe týchto informácií bola vytvorená v databáze nová tabuľka, v ktorej sa nachádzajú nasledujúce stĺpce:

- „key“,
- „hash“,
- „hostname“,
- „gcc“,
- „python“,

- „kernel“,
- „mem“.

4.2 Postup získania informácií zo servera

Na získanie informácií o serveri je použitý protokol SSH, cez ktorý sú spúšťané jednotlivé príkazy. Tento prístup je vybraný na základe jednoduchej integrácie do aplikácie. Ďalším dôvodom je predpoklad pripojenia užívateľom na server pomocou tohoto protokolu. Z toho vyplýva, že je v záujme užívateľa nastaviť správne všetky parameter skôr, ako začne používať aplikáciu. Taktiež, ak by sme chceli vytvárať nejaké spojenia medzi užívateľom a servermi, bolo by zložité vytvoriť bezpečnú komunikáciu cez internet. Preto ako najvhodnejšie riešenie je použiť protokol a aplikáciu, ktorá je overená časom ako **OpenSSH**.

Ako funguje bezpečný prenos je opísané v sekcii 2.2. Príkazy odosielané na server sa nachádzajú vo výpise 4.1.

Výpis 4.1: Použité príkazy na získanie verzií a veľkosti operačnej pamäte.

<code>gcc -dumpversion</code>	1
<code>python3 --version</code>	2
<code>uname -r</code>	3
<code>grep MemTotal /proc/meminfo awk '{print \$2 / 1024}'</code>	4

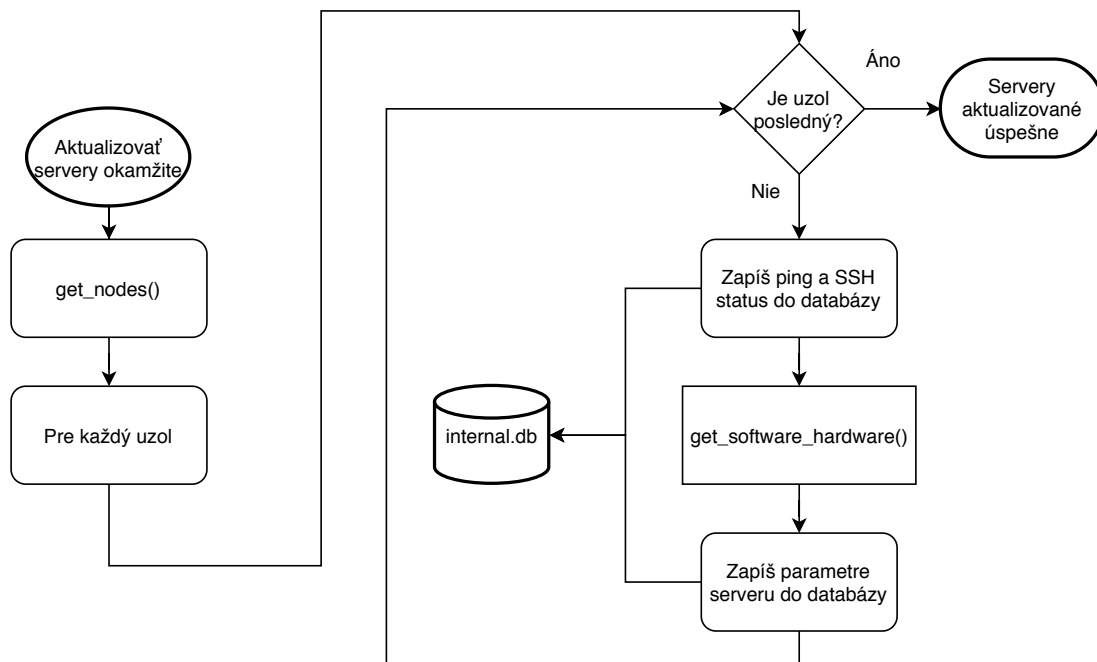
Prvé tri príkazy priamo vrátia hodnotu, ktorá sa ukladá do tabuľky. Posledný príkaz dostane hodnotu zo servera a následne lokálne túto hodnotu spracuje. Výsledok tohoto spracovania sa ukladá do tabuľky.

Je to z dôvodu, že informácia príde zo serveru v kilobajtoch, čo nie je práve veľmi čitateľná hodnota pre užívateľa. Preto je hodnota v kilobajtoch vydelená 1024 a dostaneme výsledok v megabajtoch, čo je pri dnešnej kapacite operačnej pamäte ideálna hodnota.

Taktiež, ako je opísané v kapitole 1, sú tieto servery vo väčšine prípadov len virtuálne stroje, pri ktorých nie je predpoklad, že budú disponovať veľkými hodnotami operačnej pamäte.

Vývojový diagram 4.1 znázorňuje postup pridávania informácií o serveroch do databázy. V predchádzajúcej verzii aplikácie plbmng sa zapísali do databázy len odozva na **ping**, a to konkrétne booleanovská hodnota „True“ alebo „False“. Aby riešenie bolo jednoduché, je vytvorená funkcia s názvom „get_server_params“.

Volanie funkcie je pridané k zápisu predchádzajúcich hodnôt, aby nebolo nutné upravovať už existujúcu a overenú funkcionálnosť. Funkcia vyžaduje dva vstupné pa-



Obr. 4.1: Vývojový diagram aktualizácie databázy serverov.

rametre. Konkrétne IP adresu/hostiteľský názov serveru a ako druhý parameter booleanovskú hodnotu, či má server dostupné pripojenie pomocou protokolu SSH. IP adresa alebo hostiteľský názov je vyžadovaný a nutný, pretože sa používa ako jeden z parametrov do SSH príkazu, bez ktorého nie je možné príkaz spustiť. Druhý parameter má funkciu indikovať, či je vhodné sa pokúšať o komunikáciu alebo nie. Ak server nemá otvorený štandardný port pre protokol SSH, je zbytočné pokúšať sa o komunikáciu s ním pomocou tohoto protokolu.

Toto riešenie je navrhnuté s ohľadom na úsporu času, pretože ak by bol len jeden server z tisícky bez tohoto portu, tento problém by bol zanedbateľný. Keďže ale máme k dispozícii dáta z obrázku 5.1, je navrhnuté toto riešenie, ktoré ušetrilo významné množstvo času. Príkazy z výpisu 4.1 sú uložené ako reťazec v štruktúre jazyka Python s názvom list.

Výpis 4.2: Časť kódu funkcie `get_server_params`.

```

commands = ["gcc -dumpversion",
            "python3 --version",
            "uname -r",
            "grep MemTotal /proc/meminfo \
            | awk '{print $2/1024}'",]
1
2
3
4
5
6

cmd = 'ssh -o PasswordAuthentication=no
      -o UserKnownHostsFile=/dev/null
      -o StrictHostKeyChecking=no
      -o LogLevel=QUIET
      -o ConnectTimeout=10
      -i %s %s@%s ' %
      (getSshKey(), getSshUser(), ip_or_hostname)
7
8
9
10
11
12
13

output = []
14
for command in commands:
15
    try:
16
        ret, stdout = run_command(cmd + command)
17
        if ret != 0:
18
            output.append("unknown")
19
            continue
20
        if stdout:
21
            output.append(stdout)
22
            continue
23
        else:
24
            output.append("unknown")
25

```

Vo výpise 4.2 je zobrazená najdôležitejšia časť kódu funkcie „`get_server_params`“. Podstatný je pre získanie informácií zo serveru reťazec, uložený v premennej s názvom „`cmd`“. V nej sa nachádza vopred pripravený príkaz pre terminál, v ktorom sú definované možnosti pripojenia SSH.

Tieto možnosti pripojenia sú tam definované pevne, pretože sú pre každé spustenie rovnaké. Tieto možnosti sú získané z manuálu serverovej časti **OpenSSH** s názvom `sshd` [17]:

- `PasswordAuthentication` – Parameter určuje, či pri spojení pomocou protokolu SSH je možné použiť autentizáciu pomocou hesla.
- `UserKnownHostsFile` – Špecifikuje, čo má server použiť ako databázu kľúčov známych zariadení.

- `StrictHostKeyChecking` – Určuje, ako má byť overovaný kľúč hostiteľa.
- `LogLevel` – Určuje výrečnosť pri behu príkazu.
- `ConnectTimeout` – Ukončenie spojenia v sekundách. Táto možnosť je pridaná z dôvodu úspory času.

Ďalej sa v príkaze pridáva privátny kľúč užívateľa, názov užívateľského účtu na servery a IP adresa alebo hostiteľský názov, ako bolo už spomenuté vyššie. Všetky tieto parametre sú predané ako parameter pri volaní funkcie. Potom je každý príkaz z 4.1 predaný spolu s premennou „cmd“ funkcii s názvom „run_command“. Funkcia sa nachádza vo výpise 4.3.

Výpis 4.3: Zobrazenie štandardného výstupu príkazov.

```

process = subprocess.Popen(cmd, stdout=subprocess. 1
    PIPE, stderr=subprocess.DEVNULL, shell=True)
try: 2
    stdout, stderr = process.communicate(timeout=15) 3
except subprocess.TimeoutExpired: 4
    process.kill() 5
    return process.returncode, "unknown" 6
stdout = stdout.decode('ascii', 'ignore') 7
stdout = stdout.rstrip("\n") 8
return_code = process.returncode 9
return return_code, stdout 10

```

Táto funkcia prijíma vopred pripravený príkaz ako parameter, ktorý následne spustí v potomkovi nadradeného procesu. Potomok z rodičovského procesu je vytvorený z dôvodu, že v rodičovskom procese už beží textové užívateľské prostredie. Ak nechceme užívateľovi zobrazovať na termináli zmätocné výpisy, ktoré generuje príkaz, je potrebné ich odkloniť. Toto zaisťuje vytvorenie potomka, v ktorom prebehne príkaz potichu a štandardný výstup generovaný procesom je zachytený do premennej „stdout“. Štandardný chybový výstup nezachytávame, pretože nie je pre aplikáciu nejako užitočný.

Zaistenie informácie, že zadaný príkaz nebol úspešný, prebieha pomocou premennej „retcode“. Do tejto premennej sa ukladá návratová hodnota príkazu, ktorou ukončil svoju činnosť. Na základe dokumentácie projektu **OpenSSH** [17] môžeme predpokladať, že pri návratovej hodnote „0“, všetko prebehlo v poriadku. Akonáhle návratová hodnota nadobúda akúkoľvek inú hodnotu ako „0“, aplikácia to považuje za chybu a pokračuje ďalej vo svojej činnosti. Hlbšie skúmanie chyby, pre ktorú príkaz zlyhal ďalej aplikácia nevykonáva.

Potomok rodičovského procesu je nakoniec ukončený a funkcia vracia návratovú hodnotu potomka a štandardný výstup. V štandardnom výstupe sa nachádza výstup

z príkazov vo výpise 4.1. Výstup každého z nich sa nachádza vo výpise 4.4.

Výpis 4.4: Výpis kódu funkcie `run_command`.

```
$ gcc -dumpversion 1
9 2
$ python3 --version 3
Python 3.6.9 4
$ uname -r 5
5.0.0-37-generic 6
$ grep MemTotal /proc/meminfo | awk '{print $2 / 1024}' 7
3918.66 8
```

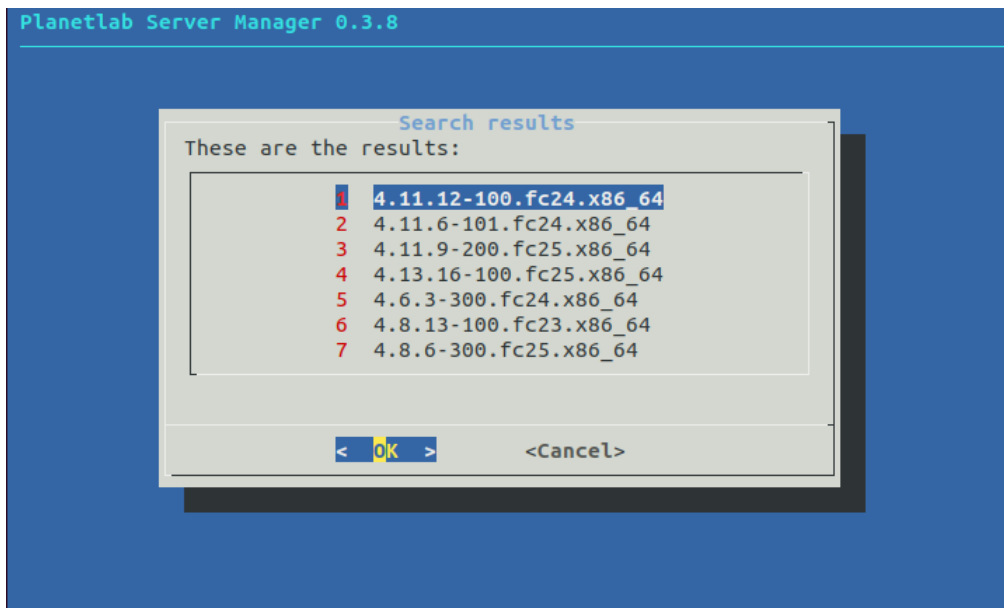
Po spustení všetkých príkazov vracia funkcia `get_software_hardware` list reťazcov podobných tým z výpisu 4.4. Tieto sú následne zapísané do databázy k danému serveru. Ak návratová hodnota niektorého z príkazov vráti inú hodnotu než nula, štandardný výstup je nahradený slovom „unknown“, ktoré vyjadruje, že hodnota nie je známa.

4.3 Využitie tabuľky s informáciami o softvéri a hardvéri

Do menu „Pristúpiť k serverom“, popísanom v sekcii 3.1, je pridaná nová možnosť „Vyhľadávať pomocou software/hardware“. Po kliknutí sa užívateľovi objavia možnosti vyhľadávania na základe príkazov spomenutých v predchádzajúcej sekcii 4.2. Po vybratí jednej z týchto možností, sa užívateľovi zobrazí verzia, prípadne množstvo pamäte. Ponuka verzií/množstva pamäte sa generuje automaticky na základe všetkých dostupných verzií/veľkosti operačnej pamäte.

Pri automatickom generovaní možností postupuje program tak, že prejde všetky dostupné servery. Pri prvom výskyte hodnoty si ju uloží a priradí k nej hostiteľský názov. Ak ktorýkoľvek ďalší server obsahuje rovnakú hodnotu, nový záznam sa nevytvára ale pridáva sa k už existujúcemu záznamu. Po prejdení všetkých serverov sa vygeneruje menu s jedinečnými výskytmi. Príklad vygenerovanej ponuky verzií jadier operačného systému Linux sa nachádzajú na obrázku 4.2.

Po vybratí jednej z týchto verzií sa užívateľovi zobrazia všetky servery s vybranou verzou. Tieto servery sú zobrazené pod hostiteľskými menami a po vybratí jedného z nich sa zobrazí okno s jeho informáciami ako na obrázku 3.3. Ďalšia možnosť využitia tabuľky bude opísaná v kapitole 5.



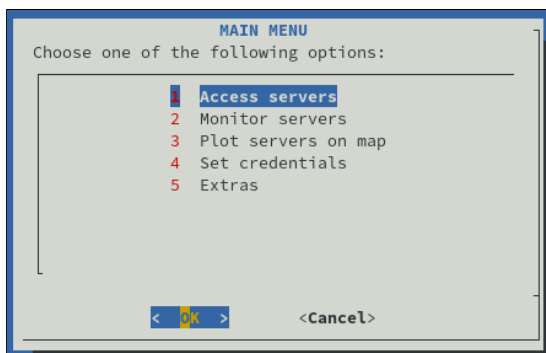
Obr. 4.2: Ukážka vygenerovanej ponuky vyhľadávania na základe verzií jadra OS Linux.

4.4 Pridávanie serverov mimo siete PlanetLab

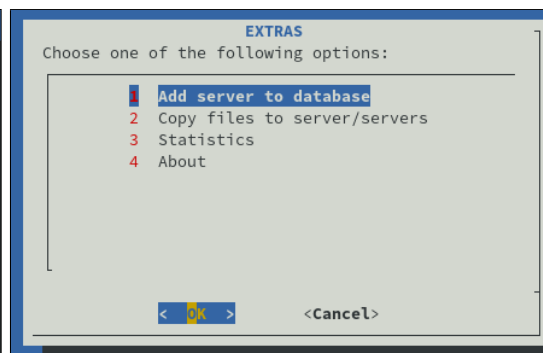
Pridanie tejto funkcionality do aplikácie plbmng bolo navrhnuté z dôvodu uľahčenia prístupu užívateľovi, ak má k dispozícii aj iné ako len servery z experimentálnej siete PlanetLab. Funkcionalita bola navrhnutá tak, aby bola čo najviac užívateľsky prívetivá a využívala podobné prvky, ako už implementované funkcie. Podobné prvky sú využívané aby zbytočne nevznikala duplicita kódu prípadne nevznikali ďalšie závislosti na externých balíkoch.

V rámci prvotných verzií bola pridaná nová možnosť s názvom „Pridať server do databázy“ do hlavného menu. Hlavné menu ale po tomto kroku začalo pôsobiť veľmi neprehľadne z dôvodu počtu možností v ňom. Preto ako riešenie tohto problému bolo zvolené vytvorenie menu, ktoré bude združovať nové rozšírenia. V rámci reštrukturalizácie bolo rozhodnuté, že sa do nového menu presunú aj možnosti „Štatistika“ a „O aplikácii“. Nová možnosť, ktorá združuje pod sebou všetky spomenuté možnosti dostala názov „Extra“. Ukážka nového hlavného menu sa nachádza na obrázku 4.3 a menu „Extra“ sa nachádza na obrázku 4.4.

Po vybratí novej možnosti „Pridať server do databázy“ sa užívateľovi zobrazí textový editor programu **dialog**, v ktorom je otvorený súbor s názvom „user_servers.node“. Súbor sa nachádza v zložke „database“, kde sa taktiež nachádzajú všetky súbory s príponou „.node“, ale aj ďalšie súbory, v ktorých sú uložené dáta aplikácie plbmng. Tento súbor je možné modifikovať akýmkoľvek iným textovým editorom podľa preferencií užívateľa. Pri pridávaní serveru sa kladie dôraz na

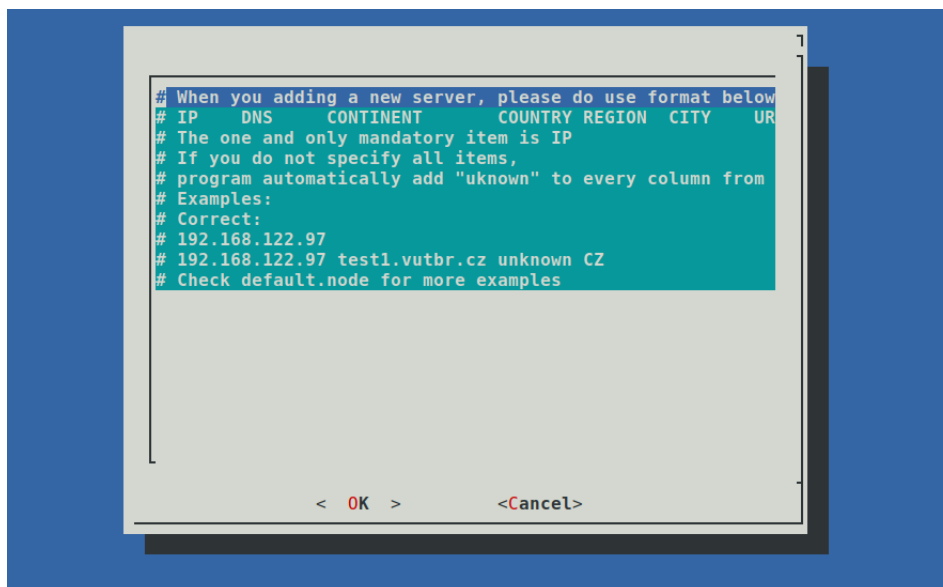


Obr. 4.3: Aktualizované menu aplikácie



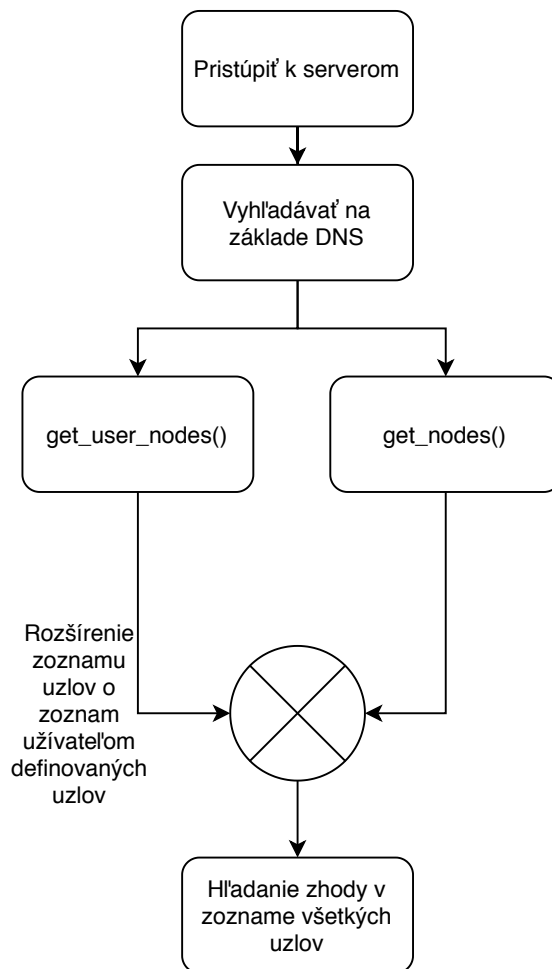
Obr. 4.4: Nové menu „Extra“

dodržanie formátu použitom pri zadávaní serveru do súboru. Pre pridanie serveru do databázy stačí zadať minimálne IP adresu. Aplikácia automaticky doplní všetky ostatné položky slovom „unknown“. V súbore sú pripravené inštrukcie s ukážkou správneho vyplnenia. Ukážka súboru sa nachádza na obrázku 4.5.



Obr. 4.5: Ukážka súboru user_servers.node v prostredí aplikácie.

Vývojový diagram 4.6 zobrazuje postup, kde bola pridaná táto nová funkcionálna do programu. Tento diagram zobrazuje prípad, keď užívateľ chce vyhľadávať v databáze server na základe doménového mena. Tento proces funguje rovnako ako v predchádzajúcej verzii, ale rozdiel je v množine, v ktorej sa zhoda hľadá. Do aplikácie bola pridaná funkcia, ktorá je volaná zároveň s predchádzajúcou a výstup týchto dvoch funkcií je spojený do jednej veľkej množiny, v ktorej následne prebieha vyhľadávanie.



Obr. 4.6: Vývojový diagram vyhľadávania serverov na základne doménového mena.

4.5 Kopírovanie súborov na servery z databázy aplikácie plbmng

Ako riešenie bezpečného kopírovania súborov bol program **scp** z projektu **OpenSSH**. Toto riešenie bolo vybrané na základe faktu, že program **scp** je vždy dodávaný spoločne s programom SSH, ako jeden kompletný balík projektu **OpenSSH**. Fakt, že SSH je hlboko implementované v tejto aplikácii, bol taktiež dôležitým faktorom pri rozhodovaní.

V dokumentácii programu **scp** je uvedená nasledovná formulácia príkazu [18]:

Výpis 4.5: Použitie programu scp získané z manuálu.

```

scp [-346BCpqrTv] [-c cipher] [-F ssh_config] [-i
  identity_file] [-J destination] [-l limit] [-o
  ssh_option] [-P port] [-S program] source ... target
  
```

Pre aplikáciu plbmng bol na základe týchto informácií vytvorený nasledujúci príkaz:

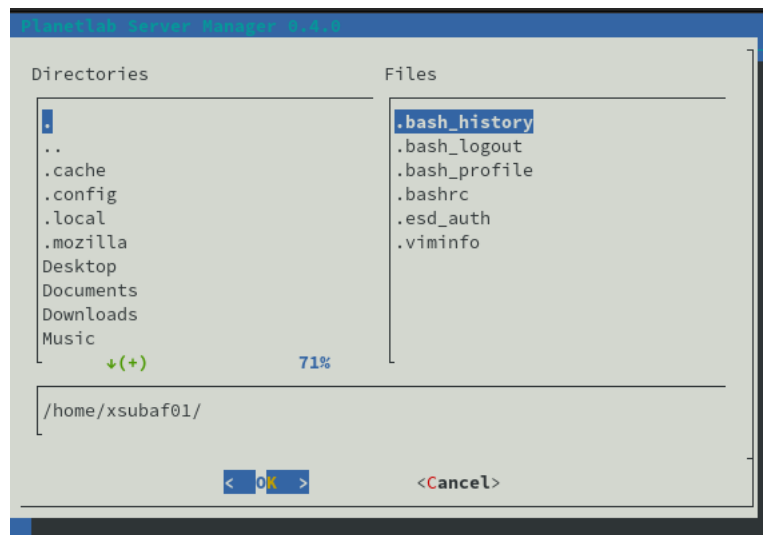
Výpis 4.6: Príkaz scp z aplikácie plbmng.

```
scp -r -o ssh_options -i identity_file source target
```

Ako premenlivé položky z výpisu 4.6 sú identifikované:

- *identity_file* (súbor s identitou) – Súbor, ktorý obsahuje súkromný kľúč užívateľa a je použitý na overenie jeho identity.
- *source* (zdroj) – Ako zdroj nebude použitý súbor, ale celá zložka, ktorá má byť prenesená na server.
- *target* (cieľ) – Ako cieľ prenosu je vždy kombinácia „užívateľ@IP_adresa/hostname:cesta“. Kde užívateľ je názov užívateľského na cieľovom serveri, IP adresa alebo hostiteľský názov cieľového serveru a nakoniec absolútna cesta k miestu, kde sa má zdrojový súbor uložiť na súborovom systéme operačného systému[18].

Aby bolo možné naplniť všetky premenlivé položky, nachádzajú sa v aplikácii pridané tri okná, ktoré užívateľ musí vyplniť. V prvom okne si interaktívne môže prechádzať svoj súborový systém, a tým pádom určí cestu k súboru alebo prípadne zložku, ktorú použijeme ako zdroj pre príkaz z 4.6. Ukážku takéhoto okna je možné vidieť na obrázku 4.7.



Obr. 4.7: Výber súboru v prostredí aplikácie.

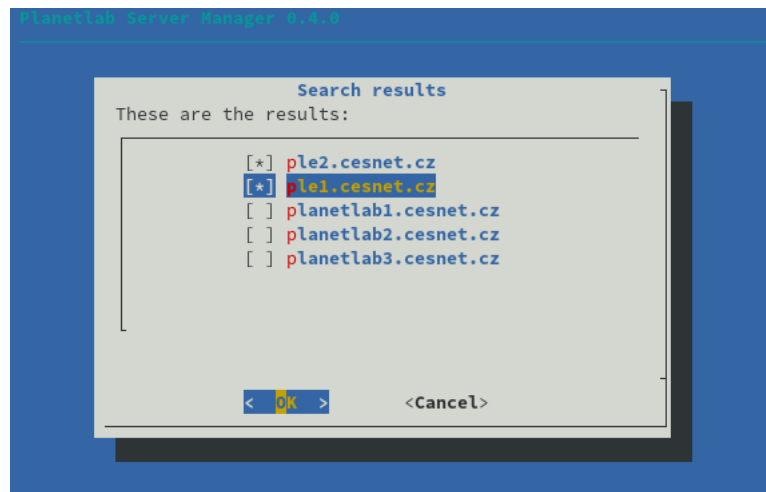
Ďalej sa zobrazí užívateľovi menu, v ktorom si vyberie server, prípadne viac serverov. Na tento účel bolo vybrané menu „Pristúpiť k serverom“. V pôvodnej forme umožňuje výber len jediného serveru, o ktorom užívateľovi zobrazí všetky

dostupné informácie. Preto bolo nutné pripraviť menu, v ktorom bolo možné zaškrtnávať viacero serverov zároveň. K tomuto účelu bola nájdená v dokumentácii projektu „pythondialog“ funkcia „checkboxlist“, ktorá umožňuje vytvárať menu s viacerými položkami na výber [19].

Výpis 4.7: Rozdiel pri použití funkcie menu a checkboxlist.

```
if not checklist:
    code, tag = self.d.menu("These are the results:",
                            choices=prepared_choices,
                            title="Search results")
else:
    code, tag = self.d.checkboxlist("These are the results:",
                                    choices=prepared_choices,
                                    title="Search results")
```

Aby bolo možné zredukovať duplicitu kódu, bolo navrhnuté riešenie pridaním nového parametru do funkcií, ktoré generujú menu pre výber serverov v „Pristúpiť k serverom“. Ak tento nový parameter nadobúda booleanovskú hodnotu „False“, funkcie pracujú v pôvodnom režime bez zmeny. Avšak pri zmene tohoto parametra na „True“ dochádza k zmene volanej funkcie na vykreslenie menu 4.8. Rozdiel medzi volaniami funkcií sa nachádza vo výpise 4.7 Po tom, čo užívateľ vyberie servery, sú



Obr. 4.8: Výber viac serverov zároveň v prostredí aplikácie.

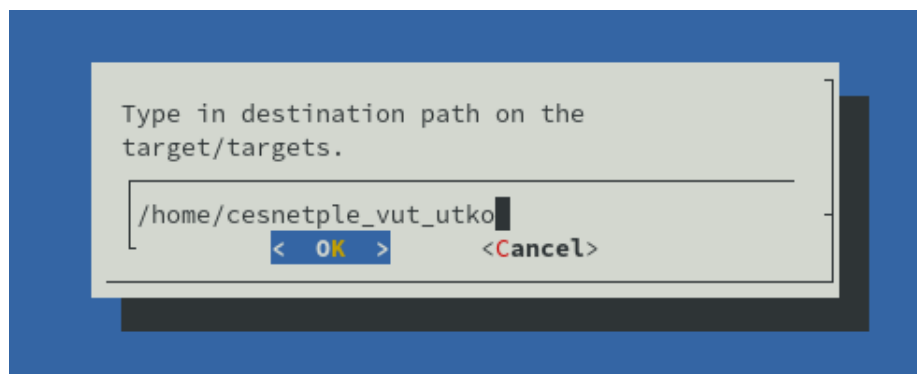
tieto vybrané servery predané ďalšej funkcii, ktorá pripravuje príkaz 4.6 do konkrétnej podoby.

Ako posledné sa užívateľovi zobrazí textové menu, kde je od neho vyžadovaná absolútna cesta na súborovom systéme cieľov. V tomto prípade by bolo takmer nemožné vytvoriť menu, v ktorom by sme mu dali na výber z ciest alebo aby bolo

možné interaktívne prechádzať po súborovom systéme na určitom ciele. Ako isté uľahčenie sa užívateľovi do políčka vyplní cesta „/home/*slice_name v konfiguračnom súbore*“. Táto cesta je štandardne domovský adresár užívateľa na súborovom systéme operačného systému Linux[8]. Na tomto základe bola táto cesta vybratá ako najpravdepodobnejší začiatok absolútnej cesty na serveri. V tomto prípade sa preto musí užívateľ spoľahnúť sám na seba a uistiť sa o správnosti cesty.

Po získaní všetkých premenlivých položiek od užívateľa, je potrebné ich spracovať. Pre tento účel bola vytvorená funkcia „parallel_copy“ a „secure_copy“. Aby bolo kopírovanie čo najrýchlejšie, sú príkazy **scp** spúšťané paralelne. Toto sa deje vo funkcii „parallel_copy“, ktorá berie ako parameter všetky vstupy od užívateľa. Táto funkcia následne spustí pre každý server funkciu „secure_copy“, do ktorej je predaný tento server ako parameter.

Kompletný príkaz je nakoniec postavený vo vnútri funkcie „secure_copy“ a je spustený. Užívateľovi sa medzitým zobrazuje okno s informáciou o priebežnom postupe kopírovania. Po ukončení všetkých spojení sa zobrazí užívateľovi oznam či sa kopírovanie podarilo. Ak nastala chyba, čo i len pri jednom prenášaní súboru, zobrazí sa užívateľovi oznam, aby overil správnosť prenesených dát na každom serveri.



Obr. 4.9: Zadanie cesty užívateľom pri kopírovaní.

4.6 Dokumentácia a refaktorizácia aplikácie plbmng

Táto časť sa bude zaoberať dokumentáciou aplikácie plbmng a všetkých jej modulov. Dokumentácia je vytvorená na základe dokumentačných reťazcov. Všetky dokumentačné reťazce sú vytvorené za účelom prehľadnosti a taktiež majú pomôcť k ľahšiemu vývoju rozšírení v budúcnosti[20]. Dokumentačné reťazce sú vytvorené pomocou značkovacieho jazyka RST[21]. Pri vytváraní bol dodržiavaný štandard kompatibilný s aplikáciou Sphinx.

Prvá veta v dokumentačnom refazci slúži na krátky popis činnosti funkcie, prípadne kedy sa používa. Následne sú pár slovami popísané všetky parametre funkcie a ich dátový typ pre lepšiu orientáciu, keďže Python určuje dátové typy automaticky a nie sú určené definíciou. Nakoniec je vo funkcii popísané čo je jej výstupom a aký dátový typ vracia.

Pre ešte lepšiu dokumentáciu sú použité prirodzené anotácie jazyka Python. Za každý parameter je doplnený jeho dátový typ a určená jeho návratová hodnota[22]. Príklad takto zdokumentovanej funkcie je možné vidieť vo výpise 4.8.

Výpis 4.8: Ukážka dokumentácie jednej z funkcií aplikácie plbmng.

```

def search_by_regex(nodes: list, option: int, regex: str) 1
    -> list:
        """
        Return all :param regex matched values from :param
            nodes at :param option index.
        """
        :param nodes: list of nodes
        :type nodes: list
        :param option: Index in the nodes list(check
            constants at the start of this file)
        :type option: str
        :param regex: Pattern to be found.
        :type regex: str
        :return: Return list of matched values as list
        :rtype: list
        """

```

Ako sa je možné dočítať v diplomovej práci Martina Kačmarčíka[16], pôvodne celú aplikáciu tvoril jediný skript. Táto skutočnosť značne komplikovala vývoj nových rozšírení, ako aj prípadnú dokumentáciu celého projektu z dôvodu neprehľadnosti. Preto bola pre ľahší vývoj vykonaná manuálna refaktorizácia zdrojového kódu aplikácie.

Refaktorizácia znamená vykonanie zmien, ktoré sú významné pre zlepšenie čitateľnosti kódu alebo predchádzajú chybám programu, ale nemenia jeho funkcionality. Pre udržanie zdravého kódu je potrebné, aby ostatní prispievatelia pridávali svoje nápady a upravovali kód do čitateľnej podoby. Pri aplikáciách so zverejneným zdrojovým kódom sa väčšinou kontroluje kód pomocou vývojárov, ktorí recenzujú a následne navrhujú zmeny pre čitateľnosť každej novej pridávanej časti kódu. Keďže aplikácia plbmng je tvorená výhradne v rámci diplomových a bakalárskych prác,

takáto možnosť chýba. Preto bola aspoň čiastočne vykonaná refaktorizácia formou revízie poslednej verzie aplikácie plbmng dostupnej na PyPi[23].

Popri vizuálnej kontrole bol pre zabezpečenie správneho štýlu Python kódu použitý program **flake8**. Flake8 preveruje, či kód, ktorý užívateľ píše, dodržiava všetky požiadavky štandardu PEP-0008. Tento štandard definuje konvencie, ktoré je nutné dodržiavať pre jednoducho čitateľný kód. Hlavné poslanstvo štandardu je konzistentnosť[24]. V definícii štandardu môžeme nájsť napríklad[25]:

- Ako odsadzovať kód.
- Či je lepšie na odsadzovanie používať tabulátor alebo medzery.
- Ako správne importovať moduly.
- Kde a koľko prázdnych riadkov treba umiestniť.

Pôvodný skript bol rozdelený na 3 súbory pre lepšiu čitateľnosť. Každý z nasledujúcich súborov obsahuje časti z pôvodného skriptu, spojené do logických celkov podľa funkcionality:

- „engine.py“,
- „database.py“,
- „library.py“.

engine.py

Tento súbor nesie pôvodný názov skriptu, z ktorého vychádza pôvodný návrh. Po vykonaní všetkých potrebných zmien sa v ňom nachádzajú už len funkcie, ktoré riadia komunikáciu s užívateľom. Na základe rozhodnutí užívateľa sú následne volané podporné funkcie nachádzajúce sa v ostatných súboroch. Pôvodný formát jednoduchého skriptu bol prerobený na triedu s názvom Engine.

database.py

Do súboru database.py boli prenesené všetky funkcie, ktoré manipulujú s internou databázou aplikácie plbmng. Tak ako v predchádzajúcom prípade bola vytvorená trieda a funkcie prerobené na metódy tejto triedy. Aby nebolo nutné komunikovať neustále s databázou, boli niektoré často používané hodnoty uložené ako atribúty triedy. Pre prípadne budúce rozšírenia aplikácie boli vytvorené metódy, ktoré neboli súčasťou projektu. Ale je pravdepodobné, že budú využité pri manipulácii s databázou.

library.py

Všetky funkcie, ktoré pracovali na spracovávaní a modifikácii dát boli presunuté do tohoto súboru. Tieto podporné funkcie nebolo vhodné presúvať do triedy z dôvodu kompatibility a nutnosti zložitejšej implementácie, ktorá nebola v rámci tejto práce časovo možná. Taktiež sa tu nachádzajú všetky funkcie, ktoré rozširujú aplikáciu plbmng.

Vygenerovanie dokumentácie

Pre vygenerovanie dokumentácie bol použitý nástroj Sphinx. Po stiahnutí nástroja je potrebné nastaviť konfiguračný súbor a pripraviť súbory, v ktorých sú definované moduly určené na vloženie do dokumentácie v jazyku RST. Ako je možné vidieť z výpisu 4.9, pre vytvorenie dokumentácie je použité rozšírenie „automodule“ [26].

Výpis 4.9: Ukážka definície modulu určeného na vloženie do dokumentácie.

```
plbmng package 1
===== 2
3
Subpackages 4
----- 5
6
.. toctree:: 7
   :maxdepth: 4 8
9
   plbmng.lib 10
11
Submodules 12
----- 13
14
plbmng.engine module 15
----- 16
17
.. automodule:: plbmng.engine 18
   :members: 19
   :undoc-members: 20
   :show-inheritance: 21
```

Toto rozšírenie je súčasťou „sphinx-autodoc“ modulu, ktorý prechádza všetky takto vopred definované moduly alebo triedy. Pri prechode modulom si zbiera všetky

komentáre vytvorené vývojárom. Po prejdení všetkých vopred definovaných moduloch je vygenerovaný výstup vo formáte, v akom ho špecifikujeme pred spustením. V tomto konkrétnom prípade bol použitý výstup vo formáte HTML. Na základe súboru ako z výpisu 4.9 sú vytvorené súbory s rovnakým názvom ale už obsahujú text dokumentácie a sú v jazyku HTML. Takto vygenerovanú dokumentáciu si je možné spustiť lokálne cez webový prehliadač po otvorení súboru „index.html“, ktorý je výstupom tohto procesu. Taktiež sa dá kompletná zložka zdieľať na webovom serveri a byť dostupná ako plnohodnotná webová stránka kedykoľvek. Z často používaných textových formátov je ešte možné použiť latex, xml, text atď.[26].

4.7 Zverejnenie aplikácie na platforme GitLab

Platforma GitLab slúži ako verejný repozitár projektov pre vývojárov aplikácií. Okrem podpory programu git na správu a kontrolu verzií poskytuje vývojárom mnoho ďalších funkcií pre efektívnejší vývoj aplikácií. Pri implementácii a refaktorizácii aplikácie plbmng bola preto použitá táto platforma. Pre účel priebežnej integrácie aplikácie plbmng, ktorú ponúka GitLab, bol vytvorený skript v jazyku YAML. Tento jednoduchý skript obsahuje úlohy, ktoré pri každom pridaní nového kódu do repozitára, platforma spustí na operačnom systéme taktiež špecifikovanom v tomto skripte. Ako je možné vidieť z výpisu 4.10, testovanie prebieha na operačnom systéme Fedora. Aby bola zabezpečená kompatibilita s najnovšou verziou operačného systému Fedora bez nutnosti prepisovania jej verzie v konfiguračnom systéme, je použitý štítok „latest“. Toto zabezpečí použitie obrazu najnovšej verzie operačného systému Fedora z centrálnej platformy **Docker**. Pod obrazom je potrebné určiť, aké programy je potrebné stiahnuť, prípadne overiť ich dostupnosť a verziu. Keďže aplikácia plbmng je napísaná v jazyku Python, ako prvé vypíšeme jeho verziu pre účely odladenia, v prípade neúspešného ukončenia testu. Ďalej sú identifikované balíky, ktoré sa nenachádzajú v obraze „fedora-latest“ a sú okamžite nainštalované pomocou programu **dnf** a **pip** [27].

Po príprave vhodného prostredia sa spustia tri úlohy:

- „test“,
- „run“,
- „pages“.

Výpis 4.10: Skript v jazyku YAML pre účel priebežnej integrácie.

```
image: fedora:latest 1
2
before_script: 3
  - python3 -V 4
  - dnf install -y which python-pip gcc python3-devel 5
    dialog
  - pip3 install setuptools cython numpy pandas 6
7
test: 8
  script: 9
    - python3 setup.py test 10
    - pip3 install flake8 11
    - flake8 --ignore=E741,W504,W503 --max-line-length 12
      =120 plbmng
13
run: 14
  script: 15
    - python3 setup.py sdist 16
  artifacts: 17
    paths: 18
      - dist/*.tar.gz 19
20
pages: 21
  stage: deploy 22
  script: 23
    - python3 setup.py install 24
    - pip3 install -U sphinx 25
    - sphinx-build -b html source public 26
  artifacts: 27
    paths: 28
      - public 29
  only: 30
    - master 31
```

Úloha „test“

Cielom tejto úlohy je zabezpečiť, že po pridaní zmien bude možné aplikáciu plbmng nainštalovať. Preto je súčasťou tejto úlohy bod, ktorý sa ju pokúsi nainštalovať. Ako ďalší bod je overenie správneho štýlu kódu tak, aby dodržiaval pravidlá PEP pomocou programu **flake8**. Tento test má zabezpečiť rovnakú čitateľnosť a kvalitu kódu, aj v prípade, že na projekte pracuje viac ľudí naraz.

Úloha „run“

Úloha „*run*“ má overiť, či je možné vytvoriť balíček aj po pridaných zmenách. Táto kontrola je veľmi dôležitá z dôvodu používania PyPi, kde sa tieto balíčky zverejňujú. Každá zmena musí zohľadňovať túto skutočnosť, aby bolo možné zachovať inštaláciu aplikácie plbmng pomocou programu **pip**. V prípade častých zmien a vydávaní nových verzií aplikácie plbmng je možné zakomponovať do tejto úlohy aj zverejnenie na PyPi.

Úloha „pages“

GitLab umožňuje zverejniť dokumentáciu prípadne iné HTML súbory ako bezplatnú webovú stránku. Každý užívateľ má túto možnosť bezplatne. Ako bolo uvedené v sekcii 4.6, bol vytvorený konfiguračný súbor a ostatné súbory, kde sú definované moduly pre automatické dokumentovanie. Vďaka tomuto je možné tento proces automatizovať a pri každej zmene vygenerovať aktualizovanú dokumentáciu [28]. Aktuálna dokumentácia je dostupná na odkaze¹.

Po vykonaní všetkých preddefinovaných úloh sa užívateľovi zobrazí stránka s výsledkom. Príklad takejto stránky s výsledkom je zobrazený na obrázku 4.10. Po kliknutí na jednotlivé úlohy je možné vidieť výpis, ako prebiehali príkazy a taktiež aj ich výstupy. Všetky tieto informácie majú prispieť k ľahšiemu odladovaniu v prípade akýchkoľvek problémov, ktoré môžu v priebehu testu nastať.

¹<https://utko-planetlab.gitlab.io/plbmng/>

passed Pipeline #148123160 triggered 3 days ago by FEEC UTKO

Add .gitlab-ci.yml

The screenshot displays the GitLab CI/CD pipeline interface. At the top, a green box indicates the pipeline is 'passed'. Below this, the pipeline details show '4 jobs for master in 3 minutes and 59 seconds'. The pipeline is triggered by the 'latest' commit, identified by the hash '73e7d26c'. A message states 'No related merge requests found.' Below the pipeline details, there are tabs for 'Pipeline', 'Jobs 4', and 'Tests'. The 'Jobs' tab is active, showing a diagram of the pipeline stages: 'Test' and 'Deploy'. The 'Test' stage contains two jobs: 'run' and 'test', both marked with green checkmarks. The 'Deploy' stage contains two jobs: 'pages' and 'pages:deploy', also marked with green checkmarks. Arrows indicate the flow from the 'Test' stage to the 'Deploy' stage.

Obr. 4.10: Ukážka úspešne otestovaného kódu z webového rozhrania GitLab.

4.8 Zverejnenie aplikácie na repozitári PyPI

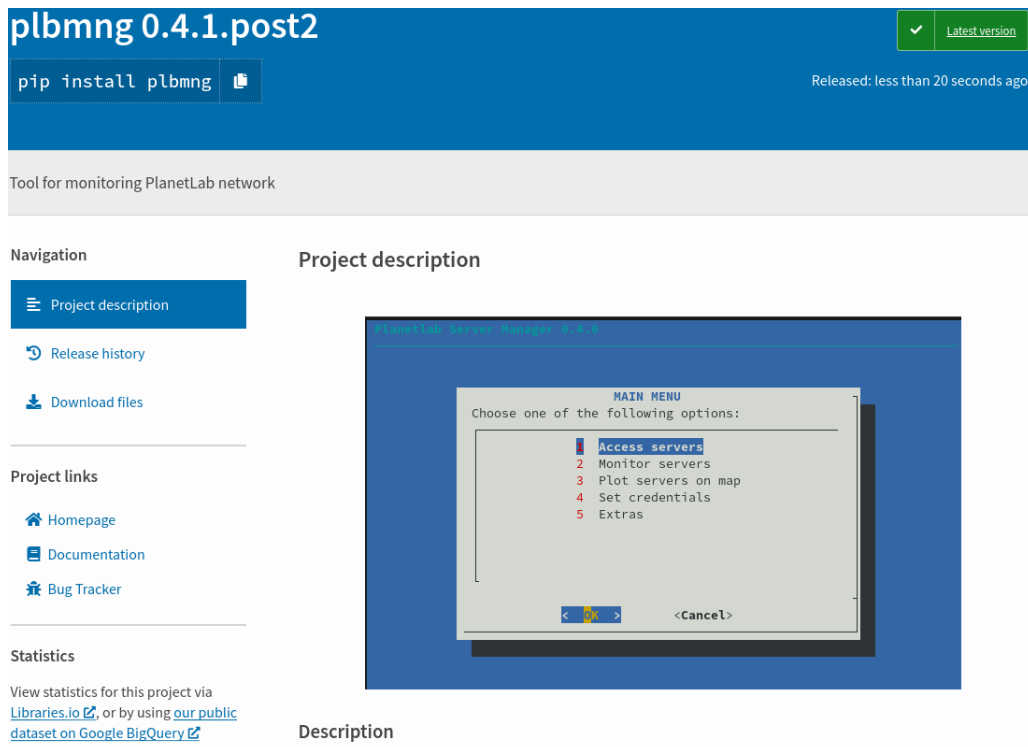
Úplne nakoniec už zostáva len zverejniť kompletný balíček s novými funkciami na repozitár PyPI. Pomocou nasledovného príkazu zobrazeného vo výpise 4.11 bol vytvorený archív so zdrojovým kódom aplikácie:

Výpis 4.11: Príkaz na vytvorenie archívu so zdrojovým kódom aplikácie.

```
python3 setup.py sdist
```

Následne bol tento nový archív zverejnený ako nová verzia. Na webovom rozhraní PyPI je možné vidieť nový aktualizovaný súbor „README.rst“. Ďalej sú pridané odkazy „Bug Tracker“ a „Documentation“. Po kliknutí na „Bug Tracker“ je užívateľ presunutý na webové rozhranie GitLab s možnosťou priamo zadávať závady. Po kliknutí na „Documentation“ je užívateľ presunutý na kompletnú dokumentáciu aplikácie plbmng, ktorá je dostupná ako webová stránka. Všetky tieto zmeny je možné vidieť na obrázku 4.11

Pomocou platformy Google BigQuery bola vytvorená tabuľka 4.1, kde sú zobrazené stiahnutia aplikácie plbmng z repozitára PyPI za posledných 6 mesiacov. Presný postup sa dajú tieto informácie získať ale aj mnohé iné informácie o štatistikách užívateľskej príručky vytvárania balíčkov jazyku Python[29].



Obr. 4.11: Ukážka projektovej stránky aplikácie vo webovom rozhraní PyPI

Tab. 4.1: Počet stiahnutí aplikácie plbmng za posledných 6 mesiacov.

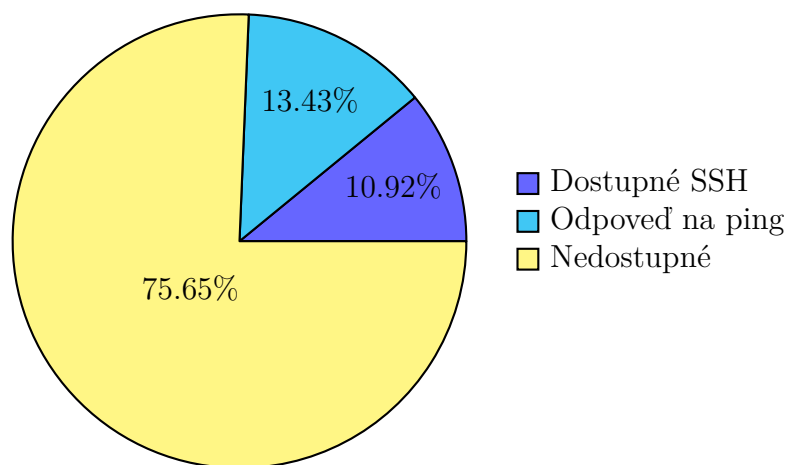
Počet stiahnutí	Kalendárny mesiac
1382	05-2020
937	04-2020
792	03-2020
963	02-2020
1029	01-2020
1620	12-2019

5 Použitie aplikácie

Kapitola sa snaží priblížiť používanie aplikácie pri správe serverov a aktuálny stav siete PlanetLab. Všetky dáta použité v nasledovnej kapitole sú získané v priebehu používania aplikácie v rámci jej testovania. V prvej časti sa kapitola zaoberá aktuálnym stavom siete PlanetLab. V nasledujúcej časti bude podrobne opísané postup použitia aplikácie.

5.1 Aktuálny stav siete PlanetLab

Na meranie aktuálneho stavu siete PlanetLab bola použitá aplikácia plbmng s už implementovanými rozšíreniami. Do tohoto momentu aplikácia poskytovala len obmedzený pohľad na servery v rámci siete PlanetLab, pretože sme mali k dispozícii len údaje ako napr. odozva na **ping** alebo dostupnosť pripojenia na port protokolu SSH. Podľa údajov z aplikácie bolo v dobe merania siete PlanetLab k dispozícii 998. Z tohoto celkového množstva serverov odpovedalo 134 na **ping**. Port protokolu SSH malo dostupných 109 serverov. V grafe 5.1 sa nachádzajú tieto hodnoty v prehľadnejšom porovnaní.



Obr. 5.1: Štatistika serverov experimentálnej siete PlanetLab vytvorená z údajov aplikácie plbmng.

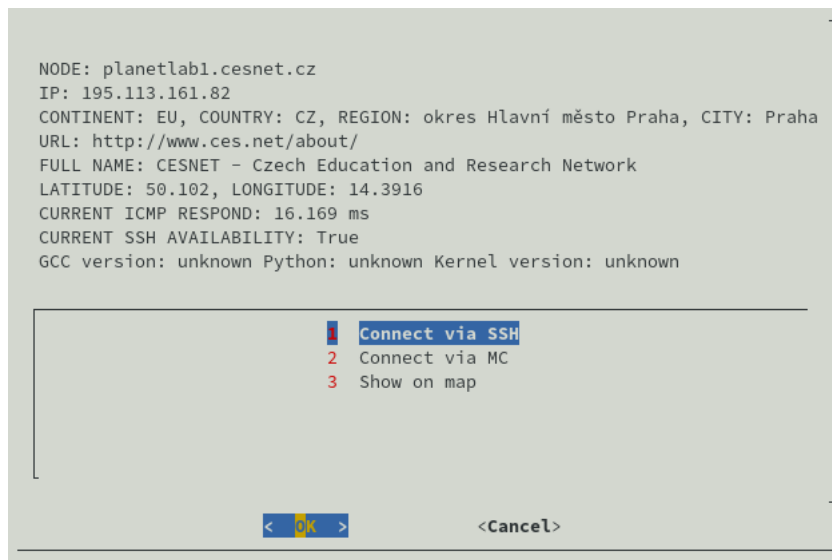
Z dostupných dát teda môžeme predpokladať odozvu s informáciami o 109 serveroch. Toto množstvo by ale bolo za ideálnych podmienok za predpokladu, že spojenie pomocou protokolu SSH prebehlo úspešne. Ako ale bolo spomenuté v predošlej časti, overuje sa len dostupnosť portu SSH s číslom 22. Z dôvodu problémov pri synchronizácii kľúčov zistenej pri testovacej vzorke, tento počet môže byť značne nižší. Toto tvrdenie môžeme dokázať chybovou správou pri pokuse o nadviazaní spojenia

a hodnotami z databázy. Pre testovanie vyberieme server z Českej republiky. Na obrázku 5.2 sa nachádzajú všetky informácie, ktoré o ňom sú dostupné v aplikácii. Ako je možné v kolónke „Aktuálna dostupnosť SSH“ vidieť, server má otvorený port 22. Avšak ak sa vyskúšame pripojiť na daný server pomocou protokolu SSH dostaneme zamietnutie pripojiť. Túto chybovú hlášku je možné vidieť vo výpise 5.1.

Výpis 5.1: Chybová správa pri pokuse o pripojení na server.

```
Warning: Permanently added 'planetlab1.cesnet.cz
,195.113.161.82' (RSA) to the list of known hosts.
cesnetple_vut_utko@planetlab1.cesnet.cz: Permission
denied (publickey).
```

Táto správa sa najčastejšie objavuje pri nastavení nedostatočných oprávnení prístupu ku súboru so súkromným kľúčom. Po overení, že sú dostatočne reštriktívne, je potrebné chybu hľadať na inom mieste. Keďže v chybovej správe sa nachádza, že prístup bol zakázaný na základe verejného kľúča, je nutné overiť funkčnosť súkromného kľúča na nejakom inom servery. Pre tento účel je vybraný iný server v Českej republike s hostiteľským menom „ple1.cesnet.cz“. Pri rovnakom spôsobe pripojenia je tento pokus úspešný. Vďaka tomuto je možné usudzovať, že kľúče sú v poriadku. Z tohto dôvodu nám zostáva už len skontrolovať priradenie serveru v diely siete PlanetLab. Pri kontrole sa v diely nachádza a preto ako posledný pokus zostáva vyradiť ho z diely a opätovne ho priradiť naspäť. Do úvahy je vzaté, že synchronizácia dielu môže trvať niekoľko dní. Preto je test pripojenia vykonaný o týždeň neskôr. Avšak aj tento pokus končí neúspechom, čím sa dostáva k bodu, kde chyba nie je na strane užívateľa ale naopak na strane poskytovateľa služby.

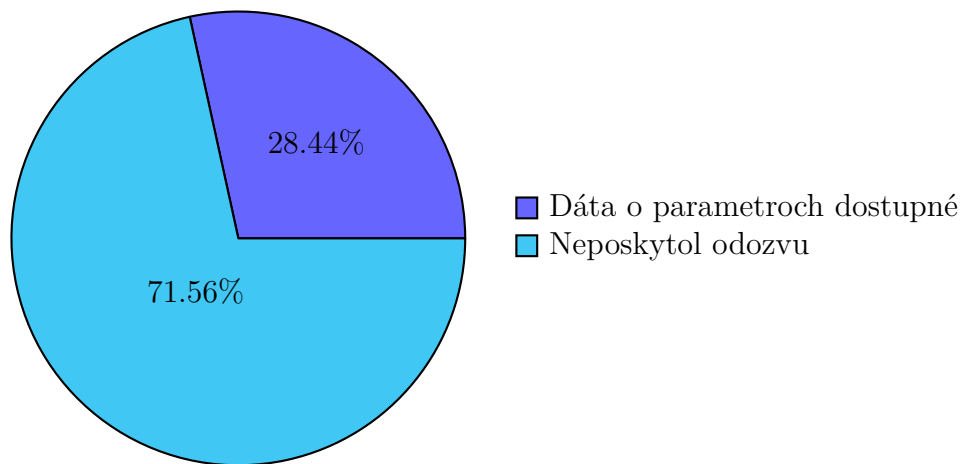


Obr. 5.2: Okno s informáciami o serveri s dostupným portom SSH.

Po naplnení databázy s informáciami o parametroch serverov bol spočítaný nasledujúcimi SQL dopytmi počet odpovedí na jednotlivé parametre vo výpise 5.2.

Výpis 5.2: Výpis SQL dopytov.

<code>select count(*) from programs where sgcc <> 'unknown';</code>	1
<code>select count(*) from programs where spython <> 'unknown';</code>	2
<code>select count(*) from programs where skernel <> 'unknown';</code>	3
<code>select count(*) from programs where smem <> 'unknown';</code>	4



Obr. 5.3: Graf zobrazujúci pomer dostupnosti informácii o serveroch.

Aký počet serverov teda je použiteľných pre vyhľadávanie, prípadne využitie? Pri spočítaní všetkých záznamov sme začínali s číslom 998. Po otestovaní odozvy na **ping** sme sa dostali na počet 137 a pri skontrolovaní portu protokolu SSH sme sa dostali na číslo 109. Nakoniec sme všetkých 109 kontaktovali a naplnili z ich odpovedí databázu. Konečný počet serverov, o ktorých máme kompletne informácie sa teda zastavil na čísle 31.

Rozbor dát

Ako bolo spomenuté na začiatku kapitoly 5, so všetkými dostupnými parametrami serverov by sme sa mohli dozvedieť reálny obraz o sieti PlanetLab. Z verzie jadra serveru je možné získať nasledovné informácie:

- Verziu jadra.
- Verziu operačného systému Fedora.
- Architektúru procesoru.

Všetky unikátne záznamy získané pomocou aplikácie v jednom čase sú prehľadne zobrazené v tabuľke 5.1.

Tab. 5.1: Verzie jadier zo všetkých dostupných serverov.

Vrátená hodnota	Verzia jadra	Verzia Fedory	Architektúra procesoru
4.6.3-300.fc24.x86_64	4.6.3-300	Fedora 24	x86_64
4.11.6-101.fc24.x86_64	4.11.6-101	Fedora 24	x86_64
4.11.12-100.fc24.x86_64	4.11.12-100	Fedora 24	x86_64
4.13.16-100.fc25.x86_64	4.13.16-100	Fedora 25	x86_64
4.11.9-200.fc25.x86_64	4.11.9-200	Fedora 25	x86_64
4.8.6-300.fc25.x86_64	4.8.6-300	Fedora 25	x86_64

Informácie o veľkosti vyrovnávacej pamäti, verzii kompilátoru GCC a programovacieho jazyka Python majú hlavne slúžiť užívateľovi pri výbere serveru. Problém nastáva pri verzii kompilátoru programovacieho jazyka C. V databáze sa nenachádza ani jeden záznam o jeho verzii. Z toho vyplýva, že je buď v aplikácii chyba alebo kompilátor sa na servery nenachádza.

Manuálnym overením bolo zistené, že kompilátor sa nenachádza v základnej verzii operačného systému Fedora. V rámci testovania bol preto nainštalovaný a opakovane bolo vykonané monitorovania stavu serverov. Po overení v databáze sa tam informácia nachádzala, a tým pádom bolo overené, že aplikácia plbmng funguje správne.

Z ostatných informácií v databáze je potom možné vidieť, že akonáhle server poskytne odozvu na verziu jadra, sú dostupné aj informácie o verzii programovacieho jazyka Python a veľkosti vyrovnávacej pamäte. Táto informácia taktiež potvrdzuje správne fungovanie aplikácie plbmng. V tabuľke 5.2 sa nachádzajú zobrazené všetky vrátené verzie jazyka Python. Taktiež sa v nej nachádzajú všetky veľkosti operačnej pamäte. To, že verzie jazyka Python sú na výber len 2, je podmienené verziami operačného systému Fedoru. V tabuľke 5.1 môžeme vidieť, že verzie operačného systému sa opakujú. Ak užívateľ nezasiahne a manuálne nenainštaluje vyššiu verziu, na 2 serveroch s OS Fedora s rovnakou verzii by mala byť rovnaká verzia programovacieho jazyka Python. V tabuľke sa nachádzajú aj všetky dostupné veľkosti operačnej pamäte. Tieto hodnoty sú očakávané, pretože ako bolo spomenuté v kapitole 1, tieto servery sú len fragmenty.

Tab. 5.2: Dostupné verzie jazyku Python a veľkosti operačnej pamäte dostupné pre vyhľadávanie v aplikácii.

Verzia jazyku Python	Veľkosť vyrovnávacej pamäte
Python 3.5.3	512M
Python 3.5.4	8192M

5.2 Príklad použitia aplikácie

V rámci tejto kapitoly bude opísaný modelový príklad použitia aplikácie. Budeme uvažovať nad situáciou, že výskumný pracovník chce overiť aké trasy sa používajú pre smerovanie z rôznych krajín Európy. Pre tento modelový príklad bude vytvorený jednoduchý skript, ktorý spustí príkaz „tracert“. Výstup príkazu sa uloží do súboru, ktorý si môže stiahnuť a uložiť pre prípadný ďalší rozbor. Ako cieľ tohto skriptu bude zvolená doména „vutbr.cz“. Tento skript sa nachádza vo výpise 5.3.

Výpis 5.3: Jednoduchý skript pre účely modelového príkladu využitia aplikácie.

```
#!/usr/bin/bash 1
2
TARGET_HOST="vutbr.cz" 3
4
tracert $TARGET_HOST > /tmp/tracert.log 5
cat /tmp/tracert.log 6
```

Nasledovný postup podrobne opisuje využitie aplikácie na modelovom príklade:

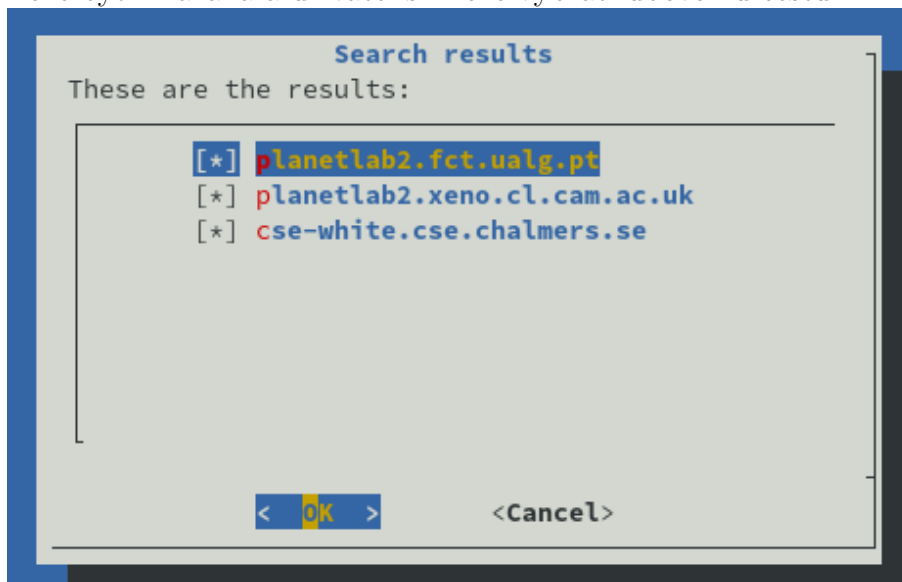
1. Užívateľ vyplní, prípadne skontroluje, všetky údaje v konfiguračnom súbore aplikácie.
2. Aktualizuje databázu pomocou možností „Okamžite aktualizovanie zoznamu serverov“ a „Aktualizovať status serverov teraz“. Užívateľ týmto získa najaktuálnejší zoznam zo siete PlanetLab a bude si môcť vyberať z aktuálne dostupných serverov.
3. Nasledne prejde do menu „Pristúpiť k serverom“. Z dostupných možností vyhľadávania si vyberie tú, ktorá vyhovuje jeho požiadavkám. V tomto modelovom príklade si vyberieme 3 náhodné servery z 3 rôznych krajín. Konkrétne sú vybrané servery z nasledovných krajín na základe dostupnosti:
 - Švédsko,
 - Portugalsko,
 - Veľká Británia.

4. Teraz je potrebné nakopírovať na vybrané servery skript, ktorý sa nachádza vo výpise 5.3. V tomto prípade je možné nakopírovať skript na každý jeden postupne pomocou programu Midnight Commander. Keďže v rámci tejto práce bola pridaná možnosť kopírovania na viac serverov zároveň, využijeme túto novú funkciu.
5. Z menu „Extra“ vyberieme funkciu „Kopírovať súbor na server“. Postupne sa objavia 3 nasledovné okná:
 - V prvom okne užívateľ vyberie cestu k súboru prípadne zložke, ktorú bude odoslaná.
 - V nasledujúcom okne vyhľadá servery a vyhovujúce označí. V tomto prípade potrebujeme nájsť 3 servery z rôznych lokácií. Z tohoto dôvodu nemôžeme použiť vyhľadávanie na základe polohy. Vyhľadávanie podľa doménového mena ale dokáže pracovať s regulárnymi výrazmi. Preto ak poznáme všetky doménové mená, môžeme si vytvoriť jednoduchý regulárny výraz vytvoriť. Konkrétne použitý regulárny výraz nájdeme vo výpise 5.4 a výstupné menu tohto vyhľadávania sa nachádza na obrázku 5.4.

Výpis 5.4: Regulárny výraz použitý na vyhľadanie konkrétnych serverov.

```
(lab2.xeno|lab2.fct|white.cse)
```

- Ako posledné používateľ zadá absolútnu cestu k miestu, kde sa má súbor uložiť. Do tejto kolónky je vopred vyplnená cesta na základe názvu dielu z konfiguračného súboru napr. „/home/cesnetple_vut_utko“. Avšak táto môže byť zmazaná a užívateľ si môže vybrať ľubovoľnú cestu.



Obr. 5.4: Výsledok vyhľadávania na základe regulárneho výrazu 5.4.

6. Následne sa spustí kopírovanie na servery a o ukončení procesu je užívateľ informovaný správou.
7. Pre následne pripojenie na jeden zo serverov je možné použiť aplikáciu plbmng alebo akékoľvek iné terminálové okno.
8. Nakoniec zostáva už len spustiť prenesený skript. Výstup zo skriptu spusteného na jednom zo serverov sa nachádza vo výpise 5.5.

Výpis 5.5: Výstup skriptu 5.3 spustenom na jednom z vybraných serverov.

```

cesnetple_vut_utko@planetlab2.xeno.cl.cam.ac.uk $ ./
tracerttest.sh
traceroute to vutbr.cz (147.229.2.90), 30 hops max,
 60 byte packets
 1  vlan293.gatwick.net.cl.cam.ac.uk
    (128.232.103.194)  3.242 ms  3.045 ms  3.677 ms
 2  cl-wgb.d-we.net.cam.ac.uk (193.60.89.1)  0.802 ms
    0.764 ms  0.560 ms
    :
 9  ae9.mx1.ams.nl.geant.net (62.40.98.128)  13.801
    ms  13.575 ms  13.542 ms
10  ae7.mx1.fra.de.geant.net (62.40.98.187)  20.347
    ms  29.721 ms  20.251 ms
11  ae8.mx1.pra.cz.geant.net (62.40.98.193)  26.950
    ms  26.914 ms  26.872 ms
12  cesnet-gw.mx1.pra.cz.geant.net (62.40.124.30)
    28.370 ms  35.653 ms  27.477 ms
13  r50-r97.cesnet.cz (195.113.157.161)  30.428 ms
    30.828 ms  30.419 ms
14  * * *
15  pe-ant.net.vutbr.cz (147.229.253.236)  37.817 ms
    37.557 ms  38.172 ms
16  irf-ant-sal.net.vutbr.cz (147.229.253.96)  37.579
    ms  39.942 ms  38.116 ms
17  piranha.ro.vutbr.cz (147.229.2.90)  29.972 ms !X
    30.003 ms !X  29.983 ms !X

```

Záver

Úlohou tejto práce bolo rozšíriť aplikáciu pre monitorovanie serverov v rámci experimentálnej siete Planetlab o nové funkcie. Tieto funkcie umožnia ľahšiu správu projektu experimentálneho výskumu v oblasti počítačových sietí v celosvetovom meradle. Za týmto účelom boli do aplikácie pridané funkcie na vyhľadávanie serverov na základe ich prevádzkových parametrov ako napríklad: veľkosť operačnej pamäte RAM, verzie jadra, verzia programovacieho jazyku Python a verzia kompilátoru programovacieho jazyku C.

Ako ďalšie vylepšenie bola pridaná možnosť pridávania vlastných serverov určených pre výskum mimo experimentálnej siete PlanetLab. Tieto servery sú potom pridané aj do vyhľadávacieho menu pre jednoduchšiu správu serverov.

Ako posledné významné rozšírenie, bolo pridané kopírovanie súborov na servery aplikácie. Aplikáciu je možné spustiť na viacerých operačných systémoch. Konkrétne je aplikácia funkčná na akejkolvek distribúcii Linux, na ktorej sú dostupné balíčky potrebné pre beh aplikácie (Python-3.5 a vyšší, dialog). Taktiež po nainštalovaní potrebných balíčkov bola aplikácia úspešne spustená na macOS Mojave(verzia 10.14.x).

V rámci práce bola taktiež zlepšená čitateľnosť pomocou rozloženia kódu z jedného súboru do viacerých, na základe logických celkov. Kód bol kompletne skontrolovaný a prepísaný, aby dodržiaval v každom možnom prípade štandard PEP-0008. K tomuto účelu bola použitá automatická kontrola pomocou nástroja Flake8.

Zdrojový kód aplikácie vrátane rozšírení bol následne zverejnený na platforme GitLab¹ pod licenciou MIT. Do projektu bol pridaný skript pre priebežnú integráciu, ktorú umožňuje platforma GitLab používať bezplatne. Pre jednoduchší vývoj aplikácie v budúcnosti bola vytvorená kompletná dokumentácia projektu s popisom ako ju používať. Následne bola vygenerovaná pomocou nástroja Sphinx webová stránka v jazyku HTML s kompletnou dokumentáciou. Na platforme GitLab bola následne zverejnená vygenerovaná dokumentácia, ktorá je dostupná z odkazu². Nakoniec bola aplikácia zverejnená na repozitáry PyPI³.

Keďže časť práce urobená v rámci semestrálneho projektu bola vydaná na repozitáry PyPI, bolo možné vytvoriť štatistiku, koľko ľudí stiahlo aplikáciu za posledných 6 mesiacov. Celkový počet stiahnutí dosiahol 6723. Z tohoto údajja môžeme zistiť, že aplikácia je žiadaná a určite má svojich aktívnych užívateľov. Podrobný popis používania aplikácie taktiež poskytuje náhľad na to, ako je možné využiť všetky rozšírenia ako jeden celok a uľahčiť jednoduchšie spravovanie serverov vedeckým pracovníkom používajúcich sieť PlanetLab.

¹<https://gitlab.com/utko-planetlab/plbmng>

²<https://utko-planetlab.gitlab.io/plbmng/>

³<https://pypi.org/project/plbmng/>

Literatúra

- [1] ANDY. Consortium. *PLANETLAB* [online]. [cit. 2019-12-21]. Dostupné z URL: <<https://www.planet-lab.org/consortium>>
- [2] About. *PLANETLAB Europe* [online]. [cit. 2019-12-20]. Dostupné z URL: <<https://www.planet-lab.eu/about>>
- [3] L. HUANG, Mark, E. FIUCZYNSKI, Marc, ed. User's Guide. *PLANETLAB* [online]. [cit. 2019-12-21]. Dostupné z URL: <<https://planet-lab.org/doc/guides/user>>
- [4] Guanlin Bian, Yuehui Jin and Tan Yang. *Delay measurement and analysis of network performance on PlanetLab*. 2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems, Shenzhen, 2014, s. 706-711, doi: 10.1109/CCIS.2014.7175825.
- [5] F. Randazzo, D. Croce, I. Tinnirello, C. Barcellona and M. L. Merani, *Experimental evaluation of privacy-preserving aggregation schemes on planetlab*. 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), Dubrovnik, 2015, s. 379-384, doi: 10.1109/IWCMC.2015.7289113.
- [6] S. Lo, M. Ammar, E. Zegura and M. Fayed, *Virtual network migration on real infrastructure: A PlanetLab case study*. 2014 IFIP Networking Conference, Trondheim, 2014, s. 1-9, doi: 10.1109/IFIPNetworking.2014.6857111.
- [7] STALLMAN, Richard. Linux and the GNU System. In: *GNU Operating System* [online]. [cit. 2019-12-21]. Dostupné z URL: <<https://www.gnu.org/gnu/linux-and-gnu.en.html>>
- [8] Linux Dokumentační projekt. 2. vyd. Computer Press, 2001. 1336 s. ISBN: 80-7226-503-2.
- [9] What is the relationship between Fedora and Red Hat Enterprise Linux? In: *RED HAT* [online]. [cit. 2019-12-21]. Dostupné z URL: <<https://www.redhat.com/en/technologies/linux-platforms/articles/relationship-between-fedora-and-rhel>>
- [10] What is Linux? In: *RED HAT* [online]. [cit. 2019-12-21]. Dostupné z URL: <<https://www.redhat.com/en/topics/linux/what-is-linux>>
- [11] Features. *OpenSSH* [online]. [cit. 2019-12-21]. Dostupné z URL: <<https://www.openssh.com/features.html>>

- [12] OpenSSH. *OpenSSH* [online]. [cit. 2019-12-21]. Dostupné z URL: <<https://www.openssh.com/>>
- [13] YLONEN, T. a C. LONVICK. RFC4251: The Secure Shell (SSH) Protocol Architecture. In: *Request for Comments* [online]. 2006 [cit. 2019-12-21]. Dostupné z URL: <<https://tools.ietf.org/html/rfc4251>>
- [14] ANDRAŠOV, Ivan: *Měření experimentální sítě PlanetLab*. Brno, 2017, 41 s. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. [cit. 2019-12-21]. Vedúci práce: doc.Ing. Dan Komosný, Ph.D
- [15] 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. [cit. 2019-12-21]. Vedúci práce: doc. Ing. Dan Komosný, Ph.D.
- [16] KAČMARČÍK, Martin. *Application for monitoring of Linux servers*. Brno, 2018, 57 s. Master's Thesis. Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications. [cit. 2019-12-21]. Advised by doc. Ing. Dan Komosný, Ph.D.
- [17] SSHD Documentation. *OpenBSD manual page server* [online]. [cit. 2019-12-21]. Dostupné z URL: <<https://man.openbsd.org/sshd.8>>
- [18] RINNE, Timo a Tatu YLONEN. SCP - OpenSSH secure file copy. *OpenBSD manual page server* [online]. 2020 [cit. 2020-05-31]. Dostupné z URL: <<https://man.openbsd.org/scp.1>>
- [19] ROUGON, Florent a Thomas DICKEY. *Pythondialog documentation* [online]. [cit. 2020-05-25]. Dostupné z URL: <<http://pythondialog.sourceforge.net/doc/index.html>>
- [20] PILGRIM, Mark. *Ponořme se do Python(u) 3*. Praha: CZ.NIC, 2010, 435 s. [cit. 2020-05-31]. ISBN 978-80-904248-2-1.
- [21] GOODGER, David. *ReStructuredText Markup Specification* [online]. Docutils Authors, 2015 [cit. 2020-05-25]. Dostupné z URL: <<https://docutils.readthedocs.io/en/sphinx-docs/ref/rst/restructuredtext.html>>
- [22] WINTER, Collin a Tony LOWNDS. PEP 3107 – Function Annotations. *Python* [online]. 2006 [cit. 2020-05-31]. Dostupné z URL: <<https://www.python.org/dev/peps/pep-3107/>>

- [23] F. Coelho, T. Massoni and E. L.G. Alves, *Refactoring-Aware Code Review: A Systematic Mapping Study*. 2019 IEEE/ACM 3rd International Workshop on Refactoring (IWor), Montreal, QC, Canada, 2019, s. 63-66, doi: 10.1109/IWoR.2019.00019. Dostupné z URL: <<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8844414&isnumber=8844408>>
- [24] STAPLETON CORDASCO, Ian. Flake8 manpage. *Flake8* [online]. c2016 [cit. 2020-05-31]. Dostupné z: <<https://flake8.pycqa.org/en/latest/manpage.html>>
- [25] VAN ROSSUM, Guido, Barry WARSAW a Nick COGHLAN. *PEP 8 – Style Guide for Python Code* [online]. In: Python. [cit. 2020-05-25]. Dostupné z URL: <<https://www.python.org/dev/peps/pep-0008/>>
- [26] BRANDL, Georg. Documentation. *Sphinx Python Documentation Generator* [online]. The Sphinx team, c2007-2020 [cit. 2020-05-25]. Dostupné z URL: <<https://www.sphinx-doc.org/en/master/>>
- [27] *GitLab Docs* [online]. San Francisco: GitLab [cit. 2020-05-31]. Dostupné z URL: <<https://docs.gitlab.com/ee/README.html>>
- [28] Gitlab Pages. *Websites for your GitLab projects, user account or group*. [online]. San Francisco: GitLab [cit. 2020-05-29]. Dostupné z URL: <<https://about.gitlab.com/stages-devops-lifecycle/pages/>>
- [29] Analyzing PyPI package downloads. *Python Packaging Authority* [online]. Python Software Foundation, c2013-2019 [cit. 2020-05-29]. Dostupné z URL: <<https://packaging.python.org/guides/analyzing-pypi-package-downloads/>>

Zoznam symbolov, veličín a skratiek

IP	Internet Protocol
SSH	The Secure Shell
HTML	Hypertext Markup Language
GNU	GNU's Not Unix
GPL	General Public License
ICMP	Internet Control Message Protocol
PyPI	Python Package Indexer
AES	Advance Encryption Standart
RSA	Rivest–Shamir–Adleman
ECDSA	Elliptic Curve Digital Signature Algorithm
CI	Continuous Integration
CD	Continuous Delivery
RST	reStructuredText
YAML	YAML Ain't Markup Language
PEP	Python Enhancement Proposals
MITM	Man-in-the-middle

Zoznam príloh

A	Obsah priloženého CD	59
A.1	Spustenie aplikácie	59
A.2	Zobrazenie dokumentácie	59

A Obsah priloženého CD

Na priloženom CD sa nachádzajú nasledujúce prílohy:

- Dokumentácia aplikácie plbmng (vrátane popisu aplikácie v anglickom jazyku).
- Zdrojový kód aplikácie plbmng.
- Elektronická verzia práce vo formáte PDF.
- README.txt – súbor obsahuje popis aplikácie v anglickom jazyku.

A.1 Spustenie aplikácie

Aplikácia bola testovaná na verzii **Python 3.8.2**. Pri inštalácii aplikácie z priloženého CD je postup nasledovný:

1. V termináli z koreňovej zložky CD sa presunúť do zložky so zdrojovým kódom.
2. V termináli spustiť príkaz **python3 setup.py install --user**, ktorý nainštaluje aplikáciu.
3. Uistiť sa, či sú stiahnuté všetky závislosti uvedené v **README.txt**, nachádzajúcim sa v koreňovej zložke CD.
4. Spustiť aplikáciu zadaním príkazu **plbmng** v termináli.

Pri inštalácii z PyPI sú kroky nasledovné:

1. V termináli spustiť príkaz **pip3 install plbmng**.
2. Spustiť aplikáciu zadaním príkazu **plbmng** v termináli.

A.2 Zobrazenie dokumentácie

Dokumentáciu je možné zobraziť dvomi spôsobmi:

- a) V obľúbenom webovom prehliadači otvoriť súbor **index.html** nachádzajúci sa v zložke **dokumentacia** v koreňovej zložke CD.
- b) Kliknutím na odkaz <https://utko-planetlab.gitlab.io/plbmng/>.