

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2016

Matej Grejták



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

EXPERIMENTÁLNÍ SÍŤ PLANETLAB

EXPERIMENTAL NETWORK PLANETLAB

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Matej Grejták

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Dan Komosný, Ph.D.

BRNO 2016

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Matej Grejták

ID: 164202

Ročník: 3

Akademický rok: 2015/16

NÁZEV TÉMATU:

Experimentální síť PlanetLab

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s experimentální sítí PlanetLab – <http://www.planet-lab.org/>. Vytvořte seznam stanic sítě PlanetLab, které se nacházejí v Evropě. Polohu těchto stanic zobrazte pomocí webové stránky. U stanic zobrazte informační box s jejich parametry - aktuální stav, IP adresa, doménové jméno a příkaz pro vzdálené připojení pomocí SSH (Secure Shell).

DOPORUČENÁ LITERATURA:

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

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

[3] CASTRO. E. HTML, XHTML a CSS : názorný průvodce tvorbou WWW stránek. 1. vyd. Computer Press, 2007. 440 s. ISBN: 978-80-251-1531-2.

Termín zadání: 1.2.2016

Termín odevzdání: 1.6.2016

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

Konzultant bakalářské práce:

doc. 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 práca sa zaoberá vizualizáciou polohy uzlov siete PlanetLab na mapovom podklade. V teoretickej časti je popísaná sieť PlanetLab, jej história, základná terminológia. Ďalej je v teoretickej časti popísané aplikačné programové rozhranie Google Maps, ktorého mapové podklady sú v tejto práci používané. Teoreticky popísané sú aj použité technológie, ako JavaScript a webový server Apache, tieto technológie tvoria, ako celok telo dynamickej webovej stránky. Praktická časť sa venuje samotnému riešeniu vizualizácie uzlov za pomoci značiek na mapovom podklade a takisto v nej sú popísané niektoré hlavné časti zdrojového kódu webovej stránky.

KĽÚČOVÉ SLOVÁ

PlanetLab, Google Maps API, JavaScript, HTML, Apache, bash

ABSTRACT

This thesis deals with visualization of PlanetLab network nodes locations defined on a base map. Theoretical part is concerned with basic knowledge of PlanetLab network, its history and general terminology. Furthermore, Google Maps application programming interface is described, which base maps are used in this thesis. Also, other used technologies as JavaScript and Apache web server are covered in the theory, these technologies combined together create the whole structure of dynamic website. Practical part is dedicated to the visualization of nodes represented by map tags. It also explains main sections of a code used on the main page with brief server configuration guide.

KEYWORDS

PlanetLab, Google Maps API, JavaScript, HTML, Apache, bash

PREHLÁSENIE

Prehlasujem, že som svoju bakalársku prácu na tému „Výzkumná síť PlanetLab“ vypracoval 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 uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorskom, o právach súvisejúcich s právom autorským a o zmene niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

BRNO

.....

podpis autora

POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce pánovi doc. Ing. Dan Komosnému, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

BRNO

.....

podpis autora



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

POĎAKOVANIE

Výskum popísaný v tejto bakalárskej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výskum a vývoj pre inovácie.

BRNO

.....

podpis autora



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	11
1 PlanetLab	12
1.1 Základná terminológia	12
1.2 Komponenty siete PlanetLab	13
1.3 Vlastnosti siete PlanetLab	14
2 Použité technológie	16
2.1 JavaScript	16
2.2 DOM – Objektový model dokumentu	18
2.3 PHP	18
2.4 Apache HTTP Server	19
3 Google Maps API	21
4 Riešenie serverovej časti	23
4.1 Skript pre získanie dát	23
4.2 Konfigurácia webového servera Apache	24
5 Riešenie vizualizácie uzlov siete PlanetLab na mape	25
6 Spracovanie a vizualizácia dát	26
6.1 Štruktúra zdrojového kódu	26
6.2 Inicializácia zdrojových dát	27
6.3 Vizualizácia značiek na mape	28
6.4 Riešenie zhodnej pozície značiek	29
6.5 Nástroj pre zobrazenie súborov LOG	31
7 Záver	33
Literatúra	34
Zoznam symbolov, veličín a skratiek	35
Zoznam príloh	36
A Obsah priloženého CD	37

ZOZNAM OBRÁZKOV

1.1	Bloková štruktúra virtuálneho servera.	13
1.2	Vizualizácia lokalít fyzických serverov PlanetLab v globálnom merítku. Prevzaté z [1].	15
2.1	Beh JavaScriptu na strane klienta.	17
2.2	Spracovanie PHP skriptu na serveri s HTML výstupom.	18
2.3	Celosvetový podiel webových serverov na aktívnych webových stránkach. Prevzaté z [10].	19
4.1	Činnosť skriptu pre prenos dát a ich roztriedenie do priečinkov.	23
5.1	Ukážka zdrojových dát z result.txt	25
6.1	Konečný výsledok po vykreslení všetkých značiek na mapový podklad.	30
6.2	Informačné okno po kliknutí na značku.	30
6.3	Výsledne riešenie zhodnej pozície značiek.	30
6.4	Zobrazenie súborov LOG.	31
6.5	Vývojový diagram behu webovej aplikácie.	32

ZOZNAM TABULIEK

6.1 Legenda pre priradenie ikon značiek na základe hodnôt SSH a PING. 29

ZOZNAM VÝPISOV KÓDU

2.1	JavaScript v kostre HTML kódu.	17
4.1	Nastavenie programu cron.	23
4.2	Obsah skriptu pre SCP prenos a roztriedenie súborov do priečinkov.	24
6.1	Volanie Google Maps JavaScript API.	26
6.2	Rozdelenie dát a presunutie do jednotlivých premenných.	27
6.3	Funkcia inicializácie značiek.	28

ÚVOD

Náplňou mojej bakalárskej práce bolo oboznámiť sa s experimentálnou sieťou PlanetLab, slúžiacou na výskum distribuovaných systémov a sieťových aplikácií v reálnom prostredí izolovanej globálnej počítačovej siete. Túto sieť, jej históriu, základné prvky a fungovanie popisujem bližšie v kapitole 1.

Mojou hlavnou úlohou bolo vytvoriť dynamickú webovú stránku vykresľujúcu uzly siete PlanetLab na mapovom podklade, technológie použité k uskutočneniu tohto cieľa popisujem v kapitole 2. V samostatnej kapitole 3 popisujem po teoretickej stránke mnou použité mapové podklady a aplikačné rozhranie Google Maps, ako celok, pomocou tohto aplikačného rozhrania vykresľujem všetky známe uzly siete PlanetLab.

Ako zdroj dát pre moju vizualizáciu uzlov siete PlanetLab používam databázu uzlov od študenta Ondřeja Pospíchala. Riešenie prenosu dát zo vzdialeného servera, roztriedenie súborov LOG, nastavenie servera a konfiguráciu webového servera Apache popisujem v kapitole 4.

Kapitola 5 je úvodom už k samotnému praktickému riešeniu zadania, teda vykresleniu značiek uzlov na mapovom podklade a takisto voľby ikony značky v závislosti od dostupnosti služieb uzla.

Posledná kapitola 6, obsahuje popísané hlavné časti zdrojového kódu mojej webovej stránky a taktiež sa zaoberá výsledkom riešenia vizualizácie.

1 PLANETLAB

PlanetLab je experimentálna globálna počítačová sieť, slúžiaca k implementácii nových komunikačných protokolov, metód distribuovaného spracovania, uchovávanía dát a ich testovania v reálnych podmienkach. Vznikla v roku 2002 na niekoľkých amerických univerzitách (University of California at Berkeley, Princeton University a University of Washington), neskôr sa k nim pridali iné univerzity z rôznych častí sveta. V súčasnosti obsahuje 1353 uzlov (nodes) na 717 lokalitách (sites) [1], celosvetovú mapu uzlov tejto siete znázorňuje Obr. 1.2. Do tejto výskumnej siete majú prístup iba určité inštitúcie, sú to predovšetkým univerzity a vývojové centrá veľkých firiem ako napr. Google, Intel a ďalšie [2].

1.1 Základná terminológia

Pre lepšie pochopenie, ako experimentálna sieť PlanetLab funguje, je potrebné ozrejmiť si základnú terminológiu a prvky tejto siete.

Site: Viaže uzly príslušnej organizácie ku geografickej polohe napr. CESNET - Czech Education and Research Network.

Node: Je to vzdialený dedikovaný server, ktorý je schopný hostiť niekoľko virtuálnych serverov. Každý uzol má svoj jedinečný názov, ktorý sa nesmie meniť. Ďalej je potrebné, aby mal uzol aspoň jednu verejnú IP adresu.

Slice: Slúži pre prístup k zdrojom siete PlanetLab, je to súbor zdrojov, poskytovaných jednotlivými uzlami PlanetLab. Ak pridáme uzol do slice tak je na tomto uzle vytvorený virtuálny server, ktorý beží po celú dobu fungovania daného uzlu v slice. Vďaka tomu sa jednotlivé experimenty navzájom neovplyvňujú. Slice je spravovaný pomocou techniky nazývanej distribuovaná virtualizácia.

Sliver: Sada alokovaných prostriedkov (súborový systém, výpočtová kapacita) na jednom uzle PlanetLab.

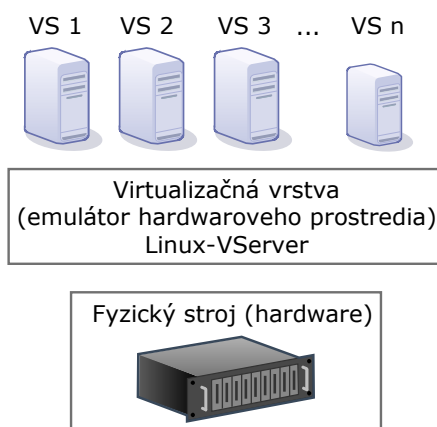
Virtual server (VS): Sliver je realizovaný ako linuxový virtuálny server (Linux-VServer). Modul prístupu k virtuálnej sieti (VNET) zabezpečuje sieťovú virtualizáciu sliver. Platí, že daný virtuálny server neregistruje sieťovú komunikáciu iných virtuálnych serverov, nemá prístup k objektom ostatných virtuálnych serverov a jeho chod nemá vplyv na výkonnosť iných virtuálnych serverov v tom istom slice. Blokovoú štruktúru virtuálneho servera môžeme vidieť na Obr. 1.1.

Principal Investigator (PI): Na danej lokalite je zodpovedný, za správu slice a užívateľov. Takisto je právne zodpovedný za všetok obsah nachádzajúci sa na slice. Väčšina lokalít má jedného PI.

Technical Contact (TC): Každá lokalita musí mať aspoň jeden technický kontakt = zodpovednú osobu, ktorá je zodpovedná za inštaláciu, údržbu a monitorovanie

uzlov prislúchajúcich k lokalite.

User: Užívateľom je každý, kto vyvíja a prevádzkuje aplikácie v sieti PlanetLab.



Obr. 1.1: Bloková štruktúra virtuálneho servera.

1.2 Komponenty siete PlanetLab

PlanetLab pozostáva z komplexného zoskupenia serverov distribuovaných po celom svete. Väčšina týchto serverov je hostovaných vo výskumných centrách a spolupracujúcich akademických inštitúciách, niektoré uzly sú situované v smerovacích centrách. Všetky tieto servery sú pripojené k internetu a tým formujú sieť, ktorá je distribuovaná, ako cez chrbticovú tak aj regionálnu sieťovú časť internetovej komunikácie [3].

Ďalej PlanetLab tvorí softvérový balíček, ktorý beží na všetkých serveroch v tejto experimentálnej sieti. Tento softvérový balíček pozostáva z týchto prvkov:

- Operačný systém založený na niektorej z Linuxových distribúcií.
- Mechanizmy pre spúšťanie uzlov a distribúciu softvérových aktualizácií.
- Kolekcia nástrojov pre manažment, monitoring, audit systémových aktivít a kontrolu systémových parametrov.
- Rozhranie pre správu užívateľských účtov a distribúciu prístupových kľúčov.

PlanetLab konzorcium je skupina akademických, firemných a vládnych inštitúcií, ktoré spolupracujú na systémovej podpore a zlepšovaní siete PlanetLab [3]. Toto konzorcium je zodpovedné za:

- Dohľadanie na rast hardvérovej infraštruktúry siete z dlhodobého hľadiska.
- Dizajn a implementáciu softvérovej architektúry.
- Poskytovanie operačnej podpory pre užívateľov siete PlanetLab.
- Definíciu politík, pre schvaľovanie projektov vytvorených v sieti.

1.3 Vlastnosti siete PlanetLab

Sieť PlanetLab bola dizajnovaná hlavne na to, aby slúžila ako testovacie prostredie pre prekrývajúce sa siete. To znamená také siete, ktoré sú vytvorené na vrchole inej siete [4]. Pre tento účel, sú akreditovaným vývojárom udelené prístupové práva k setu strojov, ktoré sú súčasťou siete PlanetLab. K správe tohto prístupu PlanetLab používa práve slice. Ak si užívateľ rezervuje nejaký slice, ihneď môže experimentovať so službami v planetárnom merítku. V súčasnosti je v sieti PlanetLab vytvorených viac než 600 projektov [1] a to z oblastí ako:

- Zdieľanie súborov, sieťové úložiska.
- CDN – Siete pre doručovanie obsahu.
- Mechanizmy detekcie sieťových anomálií.
- Škálovateľné systémy lokalizácie.
- Nástroje pre meranie sieťovej činnosti.

Všetky operácie súvisiace s manažmentom príslušného slice, sú užívateľom spracované pomocou softvérového balíka MyPLC (My PlanetLab Central). Tento softvérový balík vykonáva úlohy distribuovanej virtualizácie, ako je napríklad alokácia zdrojov sieťového hardvéru pre danú aplikáciu, alebo službu. Aplikácia bude potom bežať na jednom, alebo aj na všetkých PlanetLab serveroch, ktoré sú v danom čase dostupné a sú distribuované po celom svete. Distribuovaná virtualizácia, umožňuje mať vytvorených viacerých užívateľov a aplikácií bežiacich súčasne v rôznych slice priestoroch, vrátane prípadu ak budú rôzne priestory slice bežať na fyzicky zhodnom stroji.

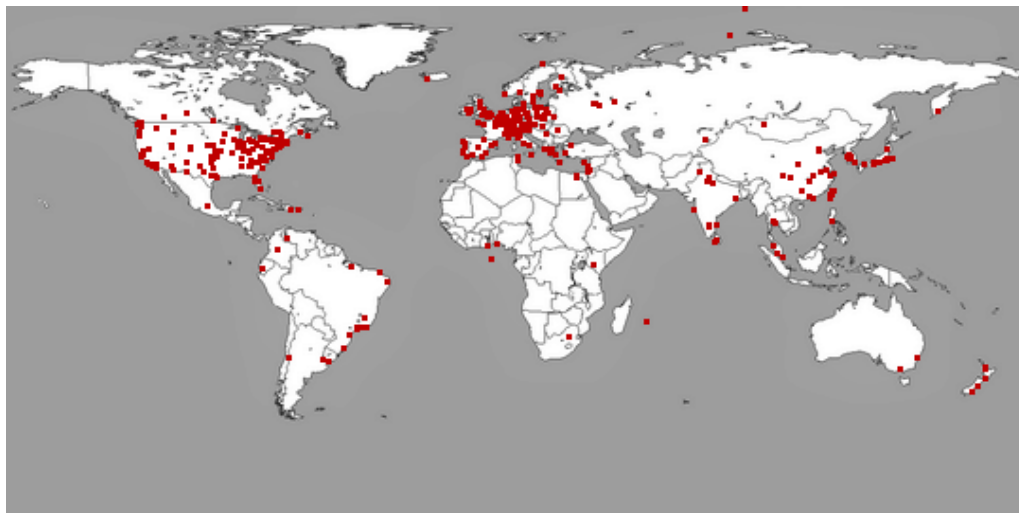
Výhoda použitia siete PlanetLab v porovnaní s experimentálnymi vývojovými sieťami, ktoré majú geograficky viazané uzly a užívateľov k jednej malej oblasti napr. univerzitné laboratórium komunikujúce len na lokálnej úrovni, spočíva obzvlášť v tom, že používatelia môžu vykonávať experimenty v reálnych podmienkach siete a predovšetkým v značne zväčšenom meradle [3]. Napríklad projekty z oblastí, ktoré boli vyššie spomenuté ťažia z distribúcie po globálnej sieti Internet, čo poskytuje vývojárom tieto benefity:

- Veľká množina geograficky rozptýlených serverov.
- Realistická sieťová infraštruktúra, zahrňajúca efekty zahltenia siete a rôznych sieťových zlyhaní pri prenose údajov.
- Realistické pracovné zaťaženie klientov, spôsobené dostatočným počtom bežiacich aplikácií na jednom PlanetLab hoste.

Príklady niekoľkých zaujímavých projektov vytvorených v sieti PlanetLab:

- OceanStore – „globálna pamäť“, dáta užívateľov sú uložené po častiach v celej sieti PlanetLab. Stránka projektu: [<https://oceanstore.cs.berkeley.edu/>](https://oceanstore.cs.berkeley.edu/).

- CoDNS – v sieti distribuovaný DNS, ktorý je odolný proti útokom a nespoľahlivosti serverov. Stránka projektu: <<http://codeen.cs.princeton.edu/>>.
- CoBlitz, CoWeb a CoDeploy – služby, ktoré využívajú distribuovanú sieť pre paralelné prenosy veľkých objemov dát [2].
- RON (Resilient Overlay Network) – tvorba a optimalizácia smerovacích tabuliek s využitím redundancie a interného monitoringu dostupných ciest v sieti. Stránka projektu: <<http://nms.csail.mit.edu/ron/>>.



Obr. 1.2: Vizualizácia lokalít fyzických serverov PlanetLab v globálnom merítku. Prevzaté z [1].

2 POUŽITÉ TECHNOLOGIE

Na strane serveru beží vo virtuálnom prostredí upravená distribúcia linuxového operačného systému Fedora a na nej webový server Apache, ktorý obsluhuje všetky HTTP požiadavky a odpovedá výstupom vo forme HTML a JavaScriptu.

2.1 JavaScript

JavaScript je multiplatformový objektovo orientovaný a interpretovaný (nie je potrebná kompilácia ako napr. v prípade jazyka C) programovací jazyk, ktorý beží na strane klienta. Beh na strane klienta znamená to, že je JavaScript spolu s HTML a ostatnými komponentami webovej stránky stiahnutý do klientskeho prehliadača, kde je následne spustený a vykonaný vid. Obr. 2.1.

Hlavným účelom JavaScriptu je tvorba DHTML (dynamického HTML). DHTML je súbor technológií a postupov pre zvýšenie užívateľského pôžitku z používania webových stránok, medzi hlavné technológie DHTML patria: HTML, JavaScript, CSS a DOM. Vývojárom umožňuje vkladať na webové stránky obsah, ktorý komunikuje s užívateľom, dynamicky generuje HTML obsah, alebo priamo riadi klientsky prehliadač. Okrem spomenutého, sa JavaScript používa tiež pre tvorbu aplikácií pre mobilné telefóny, prípadne pre tvorbu serverovo orientovaných prostredí behu ako je napríklad Node.js¹.

Syntax JavaScriptu je podobný jazykom Java a C, hoci sa v názve nachádza slovo Java, JavaScript nemá okrem z časti podobného syntaxu s Javou veľa spoločného. Pôvodne bol vyvíjaný spoločnosťou Netscape pod názvom Mocha, neskôr LiveScript. V roku 1995 bol oficiálne uvedený na trh v spolupráci s firmou Sun Microsystems a z marketingových dôvodov bolo jeho meno zmenené na aktuálne, čiže JavaScript. V roku 1997 bol štandardizovaný asociáciou ECMA (European Computer Manufacturers Association), ako ECMAScript o rok na to neskôr aj organizáciou ISO (International Organization for Standardization). Posledná 6. verzia ECMAScriptu s kódovým menom *Harmony* prijatá, ako štandard známy pod označením ECMA-262 bola vydaná v roku 2015. Každý z celosvetovo používaných internetových prehliadačov (Chrome, Mozilla Firefox atď.) podporuje verziu JavaScriptu prijatú v štandarde ECMA-262 [5]. Avšak problémom je, že každý z prehliadačov si tento štandard vykladá trochu iným spôsobom. To spôsobuje, že aj dnes sa webový vývojári používajúci JavaScript stále stretávajú s nekompatibilitou vytvoreného kódu v rôznych prehliadačoch [6].

¹Webová stránka projektu Node.js <<https://nodejs.org/>>.

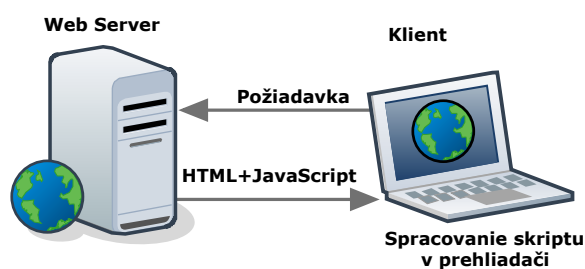
JavaScript používa dynamickú typovú kontrolu to značí, že kontrola prebieha až za behu programu a premenné nemusia mať definovaný špecifický typ, tento skriptovací jazyk rozlišuje v syntaxe taktiež malé a veľké písmená, ktoré sú častokrát zdrojom chýb v kóde [7]. Pokiaľ ide o samotnú bezpečnosť JavaScriptu, na strane klienta je v prehliadačoch znemožnený prístup k lokálnym súborom užívateľa, avšak aj to platí len z časti, kvôli možnému použitiu File API <<https://www.w3.org/TR/FileAPI/>>, obmedzený je takisto prístup k práci so sieťou. JavaScript spustený v jednom okne prehliadača pracuje len v rámci daného okna, interakcia s ďalšími aktívnymi oknami v prehliadači je blokováná, tu však takisto z časti platí výnimka a to takým spôsobom, že JavaScript dokáže iniciovať otvorenie nového ľubovoľného URL odkazu v novom okne prehliadača.

Schopnosti JavaScriptu je možné rozširovať pomocou rôznych prídavných knižníc ako sú napr.: React (pre tvorbu užívateľských rozhraní), Three.js (pre tvorbu 3D webových aplikácií), či jQuery (pre lepšiu interakciu medzi HTML a JavaScriptom). V prípade tejto práce je použitá práve knižnica jQuery.

Jazyk HTML ponúka element `script`, do ktorého môžeme vkladať spustiteľný kód jazyka JavaScript a to buď priamo napísaním kódu do časti `script` alebo odkázaním sa na súbor obsahujúci JavaScript atribútom `src` elementu `script`.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <script type="text/javascript">
5       document.write("Hello World!")
6     </script>
7   </head>
8   <body>
9 </body>
10 </html>
```

Výpis kódu 2.1: JavaScript v kostre HTML kódu.



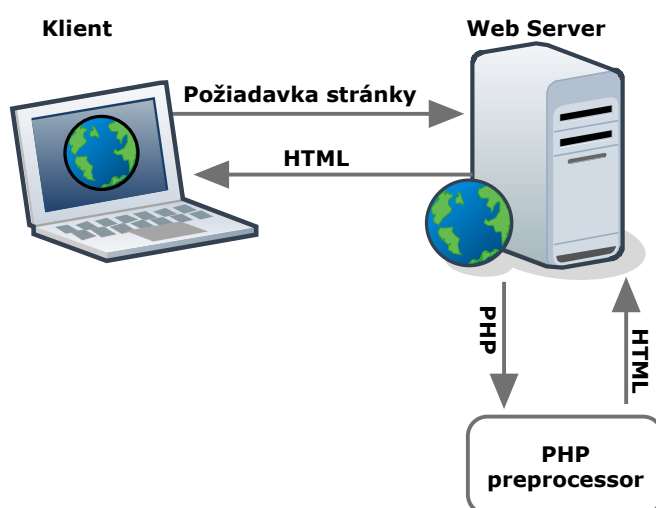
Obr. 2.1: Beh JavaScriptu na strane klienta.

2.2 DOM – Objektový model dokumentu

DOM je objektový model dokumentu vyvinutý konzorciom W3C. Je to aplikačné programové rozhranie, ktoré je nezávislé na platforme a jazyku, umožňujúce programom a skriptom dynamicky pristupovať a aktualizovať obsah, štruktúru a štýly dokumentov. To znamená, že existuje prehliadačmi používaná špecifikácia, ktorú môžu vývojári použiť k práci s webovou stránkou v dynamickom slova zmysle. DOM reprezentuje dokumenty HTML a XML vo forme stromovej štruktúry a umožňuje dynamický prístup k objektom tejto štruktúry. JavaScript je úzko spätý s modelom DOM a ponúka pre prácu s ním veľa funkcií. Avšak podobne, ako JavaScript, aj DOM interpretovali rôzne prehliadače rozdielne [6].

2.3 PHP

PHP (Personal Hypertext Preprocessor) je interpretovaný skriptovací programovací jazyk, určený pre tvorbu dynamických webových stránok. Najčastejšie sa začleňuje priamo do štruktúry jazyka HTML, XHTML či XML, čo môžeme využiť pri tvorbe webových aplikácií. Avšak PHP sa dá použiť aj k tvorbe konzolových a klientovo orientovaných aplikácií s grafickým rozhraním. Od JavaScriptu sa výrazne líši tým, že je to serverovo orientovaný jazyk a teda celý jeho beh má na starosti server, na ktorom beží takisto aj webový server. Z toho vyplýva, že klient už k zdrojovému PHP kódu nemá prístup, klient vidí už len výsledný výstup PHP skriptu, čo je častokrát už len samotný HTML kód. PHP sa používa predovšetkým v tých prípadoch, ak potrebujeme aktivitu používateľa na strane servera zaznamenávať a spracovávať [8].



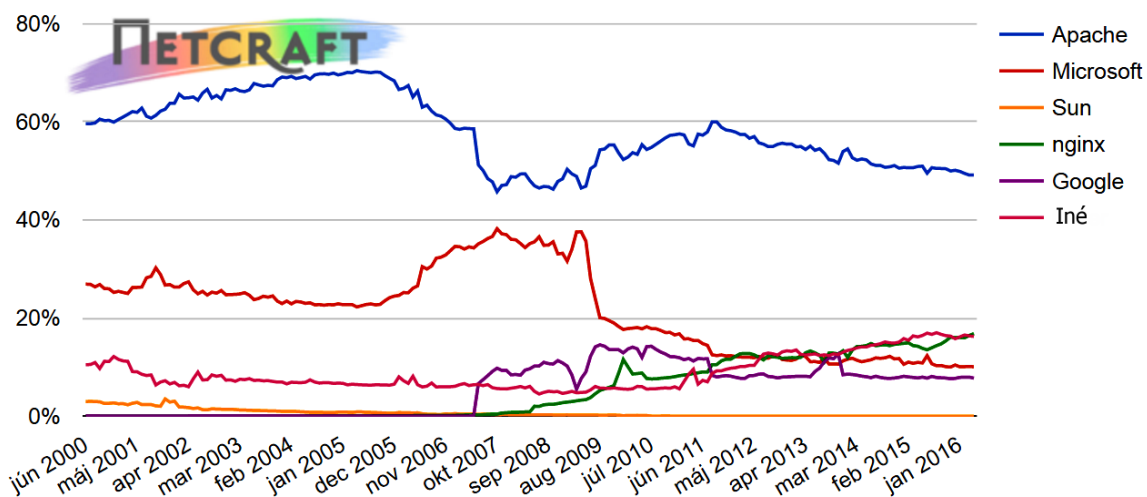
Obr. 2.2: Spracovanie PHP skriptu na serveri s HTML výstupom.

2.4 Apache HTTP Server

Apache alebo niekedy len httpd je softwarový web server. Vývoj Apache začal v roku 1995 v NCSA (National Center for Supercomputing Applications) na Illinoiskej univerzite. Pôvodné meno projektu bolo NCSA HTTPd, no Apache 2 po komplexnej úprave kódu už neobsahuje nič z pôvodného NCSA HTTPd. Súčasťou projektu Apache HTTP Server Project je aj nami použitý webový server Apache 2, avšak pod celý projekt nespadá len samotný web server, ale aj komunita nezávislých vývojárov z celého sveta, ktorí sa dobrovoľne podieľajú na ďalšom vývoji tohto projektu [9].

Webový server Apache spadá pod licenciu Apache License ver.2.0, čo umožňuje užívateľom, jeho slobodné používanie k rôznym účelom, ako distribúcií, upravovaniu, redistribúcií upravených verzií a podobne. Apache License je kompatibilná s GPL ver.3. Je multiplatformový, beží na platformách: Linux, Unix, Windows, OS X a mnoho ďalších. Medzi významnú výhodu patrí aj modularita tohto web serveru, v štandardnej základnej konfigurácii podporuje niekoľko základných vlastností, ktoré sú potrebné pre bezproblémový chod web servera. Ak používateľ potrebuje pridať nejakú novú vlastnosť, môže to dosiahnuť pomocou modulov, ktoré sú napísané v programovacom jazyku C. Napríklad pridanie podpory pre rôzne programovacie jazyky (PHP, Perl, Python, Ruby atď.), ktoré nemusia byť na stanici vôbec nainštalované, postačí len nahrať príslušného modulu.

Už od svojho vzniku patrí k najpoužívanejším web serverom na svete, k aprílu 2016 bežal na približne 82 miliónoch aktívnych stránok [10], čo je približne 50% celosvetový podiel medzi ostatnými webovými servermi vid. Obr. 2.3.



Obr. 2.3: Celosvetový podiel webových serverov na aktívnych webových stránkach. Prevezaté z [10].

Medzi hlavné výhody Apache HTTP Server patria:

- modularita.
- otvorenosť zdrojového kódu.
- výkon a spoľahivosť.
- bezplatné používanie.

3 GOOGLE MAPS API

Služba Google Maps API bola uvedená do prevádzky v roku 2005. Je to nástroj, ktorý umožňuje implementovať funkčnosť Google máp do vlastnej aplikácie. Google poskytuje niekoľko rôznych druhov API a to: Google Static Maps API, Google Maps JavaScript API, Google Street View Image API, Google Maps Embed API.

Spomínané vývojové prostredia sú určené, pre webové služby, okrem týchto API spoločnosť Google ponúka aj iné API určené špeciálne pre chod na mobilných zariadeniach bežiacich na systémoch Android a iOS. Pre bezplatné využívanie webového API konkrétne JavaScript API použitého v našej práci je nutné dodržať podmienku neprekročenia limitu 25 000 návštev za deň [11].

Typy aplikačných rozhraní Google Maps:

Google Static Maps API

Slúži pre vytvorenie a vloženie jednoduchého obrázku vybratej oblasti mapy na webovú stránku bez možnosti interakcie zo strany užívateľa. Obrázok je statický a vkladá sa na stránku len formou jednoduchého URL odkazu, čo je kladom a zároveň aj záporom tohto rozhrania. Užitočné je to, že užívateľ, ktorý chce toto API použiť sa zaobíde bez potreby použitia JavaScriptu respektíve iných skriptovacích jazykov. Hlavným negatívom tohto API je už spomínaná absencia dynamických prvkov ako napr. priblíženie a oddialenie mapy. Z toho dôvodu je použitie Static Maps API pre naše zadanie vysoko nepraktické a neefektívne.

Google Street View Image API

Podobne ako u predchádzajúceho Static Maps API ide o vloženie obrázku na webovú stránku. Avšak v tomto prípade už nejde o vkladanie vybranej mapovej oblasti, ale o vloženie užívateľom zvoleného pohľadu v Google Street View (Google Street View je služba, ktorá ponúka 360-stupňový panoramatický pohľad na ulice a ich okolie). Parametre pohľadu sú definované na základe jedinečných parametrov nachádzajúcich sa v URL adrese, ktorá je odoslaná ako štandardná HTTP požiadavka na základe ktorej je späť vrátený spomínaný obrázok.

Google Maps Embed API

Použitím sa veľmi podobá Google Static Maps API, avšak toto rozhranie už nie je statické, ale dynamické, čo umožňuje pohyb na mape aj mimo zvoleného pôvodného výberu mapy a takisto aj priblíženie a oddialenie mapy. Voľba výberu mapy

a jeho parametre sú opäť uskutočnené pomocou jednoduchého HTTP požiadavku definovaného v URL adrese, tak ako v predchádzajúcich dvoch príkladoch.

Google Maps JavaScript API

Umožňuje vkladať mapy do webového prostredia pomocou JavaScriptu. V súčasnosti je najaktuálnejšia jeho 3. verzia predstavená v roku 2009, ktorá na rozdiel od tej predchádzajúcej využíva asynchrónne nahrávanie kódu, čo enormne znížilo čas potrebný pre spracovanie mapového kódu. Vďaka podpore JavaScriptu je toto vývojové rozhranie schopné vytvárať dynamické interaktívne mapy. Takže použitím rôznych funkcií sme schopný si toto mapové prostredie voliteľne modifikovať a to napr. voľbu mapového podkladu, celkový design prostredia, či chovanie ovládacích prvkov. V mojej práci využívam práve JavaScript API a v ňom pridávanie nových objektov na mapový podklad, konkrétne značiek.

4 RIEŠENIE SERVEROVEJ ČASTI

Použil som linuxový server, ktorý beží v rámci uzlu siete Planetlab, ktorého hardware sa nachádza na Technickej univerzite v Stuttgarte. Všetky komponenty spomínané v tejto kapitole sú realizované na tomto serveri.

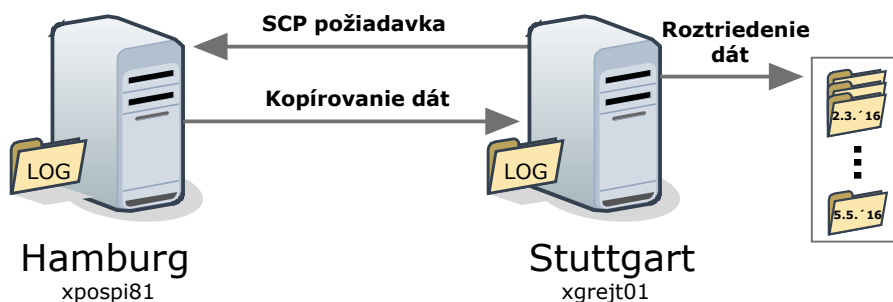
4.1 Skript pre získanie dát

Na serveri používam program `cron`. Tento program spúšťa v pravidelných intervaloch každý deň, vždy o 23:30 skript `LOG_sync_scp.sh`. Výstup skriptu je presmerovaný do súboru `cron_log`, ktorý slúži, ako logovací súbor pre kontrolu funkcionality spúšťaného skriptu.

```
1 30 23 * * * /home/cesnet_feec/xgrejt01/LOG_sync_scp.sh >> /home/cesnet_feec/xgrejt01/cron_log 2>&1
```

Výpis kódu 4.1: Nastavenie programu cron.

Samotný skript, ktorého obsah je na výpise kódu 4.2, má za úlohu predovšetkým zabezpečený prenos zdrojových dát pre moju webovú stránku pomocou protokolu SCP. Dáta sú kopírované zo vzdialeného serveru v Hamburgu, na ktorom beží program vytvorený Ondřejom Pospíchalom. Skript je ošetrený pre prípadnú chvíľkovú nedostupnosť servera opakovaním prenosu s obmedzeným počtom pokusov a oneskorením 20 minút medzi jednotlivými pokusmi. Po úspešnom prenose sú pre väčšiu prehľadnosť dáta rozdelené do novovytvorených priečinkov s názvom odpovedajúcim dátumom vytvorenia nachádzajúcim sa v názvoch súborov. Grafické znázornenie fungovania skriptu viď. Obr. 4.1



Obr. 4.1: Činnosť skriptu pre prenos dát a ich roztriedenie do priečinkov.

```

1 n=0
2   until [ $n -ge 10 ]
3   do
4       scp -C -i /home/cesnet_feec/xgrejt01/id_rsa_open -r
        cesnet_feec@merkur.planetlab.haw-hamburg.de:/home/
        cesnet_feec/xpospi81/"Result.txt /home/cesnet_feec/
        xpospi81/planetlab.node /home/cesnet_feec/xpospi81/LOG/* "
        /var/www/LOG && break
5       n=$((n+1))
6       sleep 600
7   done
8 d=/var/www/LOG
9 for i in `ls $d`
10 do
11     DIR=`echo $i | cut -d '_' -f 2`
12     if [ ! -d "$d/$DIR" ]; then
13         mkdir $d/$DIR
14     fi
15     mv $d/$i $d/$DIR/$i
16 done

```

Výpis kódu 4.2: Obsah skriptu pre SCP prenos a roztriedenie súborov do priečinkov.

4.2 Konfigurácia webového servera Apache

Na serveri bola nainštalovaná najnovšia možná verzia webového serveru Apache, ktorú podporuje operačný systém Fedora 14 a to príkazom:

```

1 sudo yum install -y httpd

```

Ďalej bola spustená služba `httpd` a doinštalovaný prídavný modul pre podporu jazyka `php`. Pre špecifické potreby môjho riešenia som editáciou konfiguračného súboru `httpd.conf` nastavil potrebné parametre servera a otvoril port s číslom 7005 k prístupu na moju webovú stránku.

5 RIEŠENIE VIZUALIZÁCIE UZLOV SIETE PLANETLAB NA MAPE

Moje riešenie zadania tejto záverečnej práce spočíva vo vytvorenej dynamickej webovej stránke, ktorá je verejne dostupná na adrese <<http://planetvs2.informatik.uni-stuttgart.de:7005/>> vďaka webovému serveru, ktorý beží na jednom z uzlov siete Planetlab. Zdrojový kód tejto webovej stránky sa z väčšej časti skladá zo skriptu napísaného v skriptovacom jazyku JavaScript. Stránka takisto využíva PHP a vývojové prostredie Google Maps JavaScript API.

Všetky potrebné časti stránky sú zakomponované v troch zdrojových súboroch stránky a to:

- `index.html`.
- `oms.min.js`.
- `dir_index.php`.

Dáta, ktoré slúžia k načítaniu polohy značiek na mape sú obsiahnuté v súbore `result.txt`. Tieto dáta mi sprostredkoval Ondřej Pospíchal, ktorý sa zaoberal práve problematikou získania potrebných dát testovaním jednotlivých uzlov siete PlanetLab.

```
roti.mimuw.edu.pl 193.0.109.23 ping 0 ssh 1 52.211 20.981
planetlab1.cs.uoi.gr 195.130.121.204 ping 1 ssh 1 39.6182 20.8386
planetlab2.mini.pw.edu.pl 194.29.178.6 ping 0 ssh 1 52.2221 21.0069
planetlab03.cnds.unibe.ch 130.92.70.253 ping 1 ssh 1 46.953594 7.439691
planet1.itc.auth.gr 155.207.48.21 ping 1 ssh 1 40.633569 22.956843
```

Obr. 5.1: Ukážka zdrojových dát z `result.txt`

Dátový súbor je delený po riadkoch, kde sa na každom riadku nachádzajú príslušné hodnoty parametrov pre každý jedinečný uzol v sieti PlanetLab. Hodnoty parametrov sú v riadkoch oddelené tabulátorom, pre jednoduchosť ďalšieho spracovania dát. Každý riadok obsahuje DNS názov uzlu, jeho IP adresu, hodnotu PING, ktorá určuje dostupnosť uzlu (1 nedostupný, 0 dostupný), dostupnosť SSH spojenia v tom istom formáte ako je tomu pri parametre PING a na koniec zemepisné súradnice polohy uzlu v podobe zemepisnej šírky a zemepisnej dĺžky lokality miesta na Zemi.

6 SPRACOVANIE A VIZUALIZÁCIA DÁT

Moja dynamická webová stránka používa mapové podklady od spoločnosti Google. Vizualizácia je riešená pomocou Google Maps Javascript API, ktoré bližšie popisujeme v kapitole 3.

6.1 Štruktúra zdrojového kódu

Kostru kódu webovej stránky tvoria prevažne mapové funkcie a metódy Google Maps JavaScript API písané v JavaScripte. To nám značne zjednodušilo výsledné spracovanie práce, pretože mnoho nástrojov je už implementovaných priamo v API.

V predchádzajúcej verzii Google Maps JavaScript API V2 bolo nutné k sfunkčneniu prostredia mať vytvorený účet Google a vygenerovaný API kľúč. Túto požiadavku ošetrila spoločnosť Google v najnovšej verzii API V3 a teda pre používanie tohto prostredia musíme len vhodne umiestniť potrebnú časť kódu volajúcu API do nášho zdrojového kódu stránky [12].

```
1 <script type="text/javascript">
2 src="https://maps.googleapis.com/maps/api/js?v=3.exp&callback=
   initMap">
3 </script>
```

Výpis kódu 6.1: Volanie Google Maps JavaScript API.

Inicializácia mapového podkladu príkazom:

```
1 var map = new google.maps.Map(document.getElementById('map'),
```

Pri tejto inicializácii je možné nastaviť defaultné hodnoty mapového prostredia a jeho prvkov, ktoré sa budú zobrazovať na mape. V mojom prípade nastavujem hodnotu priblíženia mapy, tak aby bola z počiatku viditeľná takmer celá oblasť, kde budú vysádzané značky, K centrovaniu mapy som použil zemepisné súradnice 25°48'00.5"N 10°27'14.4"E. Rozloženie ovládacích prvkov mapy, ako tlačidlá pre priblíženie, oddialenie, prepnutie módu mapy zo zobrazenia terénu na satelitné snímky, riešim z toho dôvodu, že mi mnou navrhnuté rozloženie prišlo pre moje zadanie užívateľsky prívetivejšie, než to defaultné od autora API. Tieto úkony som vykonal pomocou príkazu `position`, vid. príklad:

```
1 zoomControl: true,
2 zoomControlOptions: {
3   position: google.maps.ControlPosition.LEFT_CENTER },
```

6.2 Inicializácia zdrojových dát

Po načítaní základných komponentov stránky, ako je legenda mapy, horný panel stránky obsahujúci ovládacie prvky nasleduje vykreslenie mapového podkladu. Ihneď potom dochádza k načítaniu zdrojového súboru `result.txt` pomocou príkazu:

```
1 $.get('../LOG/result.txt')
```

Príkaz `$.get` pochádza z JavaScriptovej knižnice jQuery a zabezpečí načítanie dát priamo zo servera pomocou požiadavky HTTP GET[13].

Po načítaní zdrojového súboru, prehliadač pokračuje v rozdelení dát a presunie ich do na to určených premenných. Celý tento proces prebieha tak, že skript načítaný dátový súbor tvorený z mnohých riadkov rozdelí na jednotlivé samostatné riadky. Otestuje, či sa v danom riadku nenachádza symbol, ako napríklad mriežka `#`, takéto riadky zo zoznamu vylúči a následne validné riadky, v ktorých je každý parameter oddelený tabulátorom nahraje do špecifických premenných funkcie `serverInfo`.

```
1 function parseData(file_content) {
2   var data = file_content.split('\n').filter(function(row) {
3     return !/^\s*?#/\.test(row);
4   }).map(function(row) {
5     return row.trim().split('\t');
6   });
7   var version = data.splice(0, 1).toString();
8   return data.filter(function(info) {
9     return info.length > 1;
10  }).map(function(info) {
11    return {
12      hostname: info[0],
13      ip: info[1],
14      ping: info[2],
15      ping_status: info[3],
16      ssh: info[4],
17      ssh_status: info[5],
18      latitude: info[6],
```

```

19         longitude: info[7],
20     };
21 });
22 }

```

Výpis kódu 6.2: Rozdelenie dát a presunutie do jednotlivých premenných.

6.3 Vizualizácia značiek na mape

Keďže, sú už všetky dáta v potrebných premenných nastupuje hlavná časť mojej záverečnej práce a to zobrazenie značiek označujúcich polohu uzlov siete PlanetLab na mapovom podklade. Pomocou funkcie `initMarkers` sa inicializujú značky, no ešte sa nevykresľujú. Značkám sa v tejto funkcii nastaví parametre, ktoré určujú objekt, na ktorom sa budú vykresľovať a ich ďalšie vlastnosti, ako v našom prípade, čo sa stane, ak užívateľ klikne na zobrazenú značku.

```

1  function initMarkers(map, servers) {
2      var oms = new OverlappingMarkerSpiderfier(map);
3      oms.addListener('click', function(marker) {
4          var server = servers.find(function(server) {
5              return server.marker === marker;
6          });
7          if (server) {
8              server.infoWindow.open(map, marker);
9          }
10         openedServer = server;
11     });
12     servers.forEach(function(server) {
13         server.marker.setMap(map);
14         oms.addMarker(server.marker);
15     });
16 }





```

Výpis kódu 6.3: Funkcia inicializácie značiek.

Z kódu vyplýva, že po kliknutí na značku je aktivovaný `Listener`, ktorý zobrazí informačné okno `server.infoWindow` obsahujúce parametre daného uzla.

Funkcia `MarkerColorChanger` s parametrami `serverInfo.ping_status`, `serverInfo.ssh_status`, nám na základe hodnôt: 1-dostupný, 0-nedostupný, uložených v premenných `ping_status` a `ssh_status`. Určí aká ikona značky má byť priradená danému uzlu, priradenie ikon sa nachádza v tabuľke 6.1.

Tab. 6.1: Legenda pre priradenie ikon značiek na základe hodnôt SSH a PING.

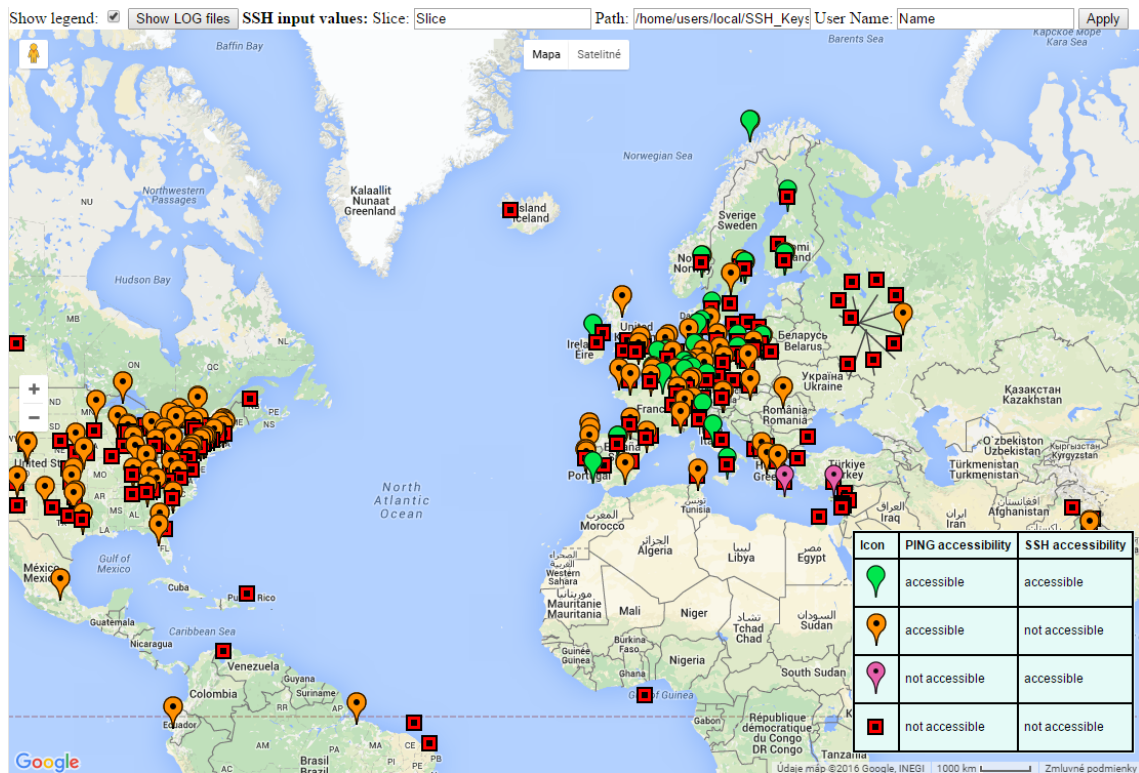
Ikona	Dostupnosť SSH	Dostupnosť PING
	dostupné	dostupné
	dostupné	nedostupné
	nedostupné	dostupné
	nedostupné	nedostupné

Metóda `return new google.maps.Marker`, až na záver zabezpečí samotné vykreslenie značiek do určenej lokality po prečítaní ich zemepisnej šírky a dĺžky. Táto metóda navyše obsahuje aj parameter, ktorý volá už zvolenú ikonu značky a takisto volá funkciu `google.maps.Animation.DROP` vytvárajúcu animáciu pri počiatočnom vykreslení značiek, animácia simuluje efekt tzv. zhodenia značiek na mapu. Horný panel stránky sa skladá z ovládacích prvkov užívateľského rozhrania webovej stránky. Obsahuje zaškrtačacie pole pre zobrazenie, alebo skrytie legendy, tlačidlo, ktoré užívateľa presmeruje do sekcie stránky pre zobrazenie súborov z priečinka LOG a takisto textové polia, ktoré je možné editovať a tým pádom zmeniť obsah syntaxu v textovom poli informačného okna pre pripojenie k SSH vid. Obr. 6.2, po kliknutí na užívateľom zvolenú značku uzlu siete PlanetLab na mapovom podklade.

6.4 Riešenie zhodnej pozície značiek

Po zistení, že mnoho uzlov siete PlanetLab má v údajoch o svojej fyzickej polohe zaznamenané zhodné hodnoty súradníc zemepisnej šírky, zemepisnej dĺžky a teda by pri vykreslení došlo k prekryvaniu viacerých značiek, bolo nutné daný problém ošetriť. Vzhľadom na to, že tzv. zhukovanie značiek obsiahnuté v samotnom Google Maps API v3 daný problém nerieši, rozhodol som sa ísť cestou doplnku OMS (*Overlapping Marker Spiderfier*)¹. Kód tohto doplnku po kliknutí na skupinu prekryvajúcich sa značiek rozloží značky do kruhu, resp. v prípade väčšieho počtu prekryvajúcich sa značiek do špirály pozostávajúcej z lúčov, na ktorých konci sú jednotlivé značky vid. Obr. 6.3 . Samotná integrácia doplnku spočívala v úprave značnej časti kódu obsluhujúcej mapový podklad a jeho prvky.

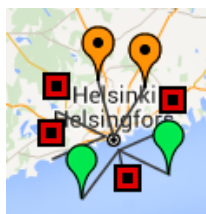
¹Zdrojový kód je dostupný z <<https://github.com/jawj/OverlappingMarkerSpiderfier>>.



Obr. 6.1: Konečný výsledok po vykreslení všetkých značiek na mapový podklad.



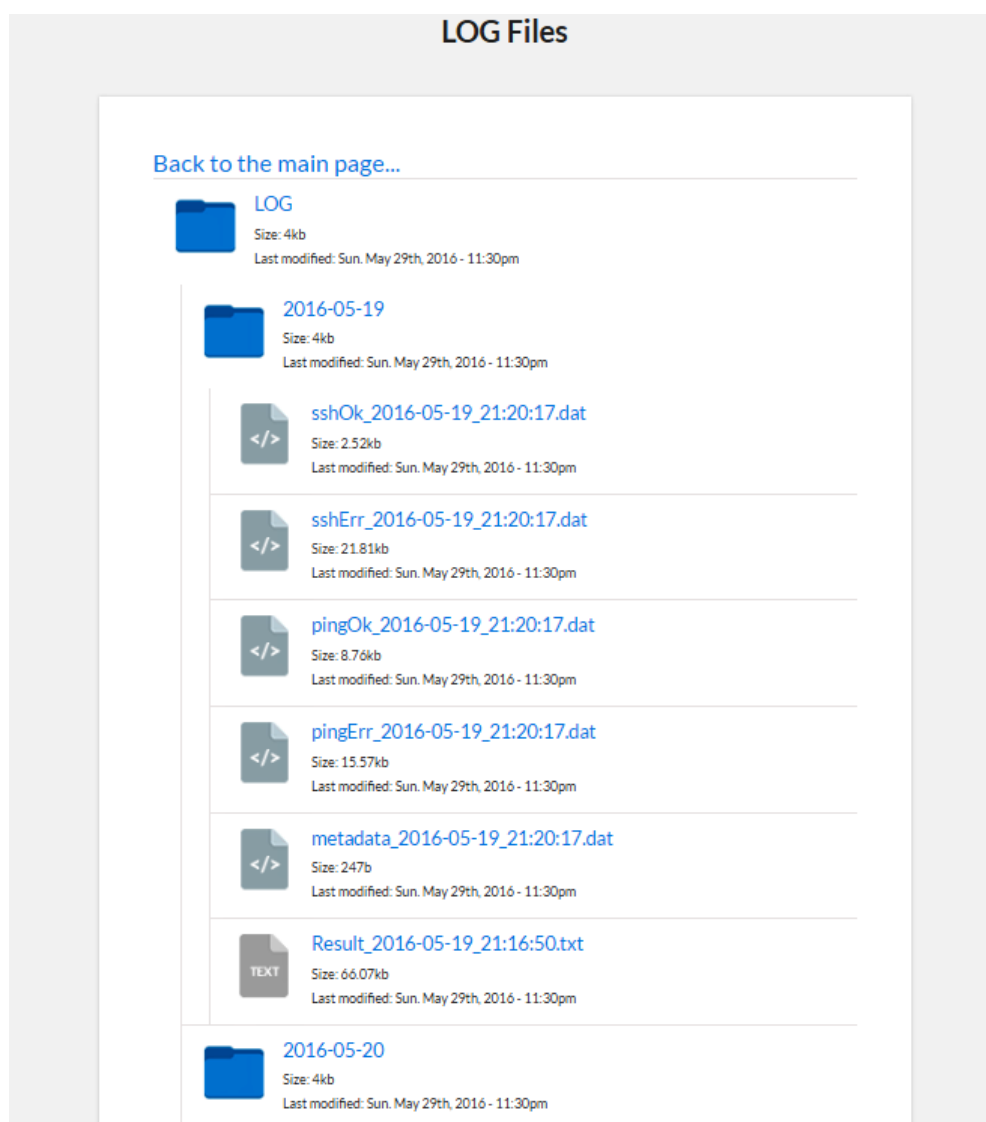
Obr. 6.2: Informačné okno po kliknutí na značku.



Obr. 6.3: Výsledne riešenie zhodnej pozície značiek.

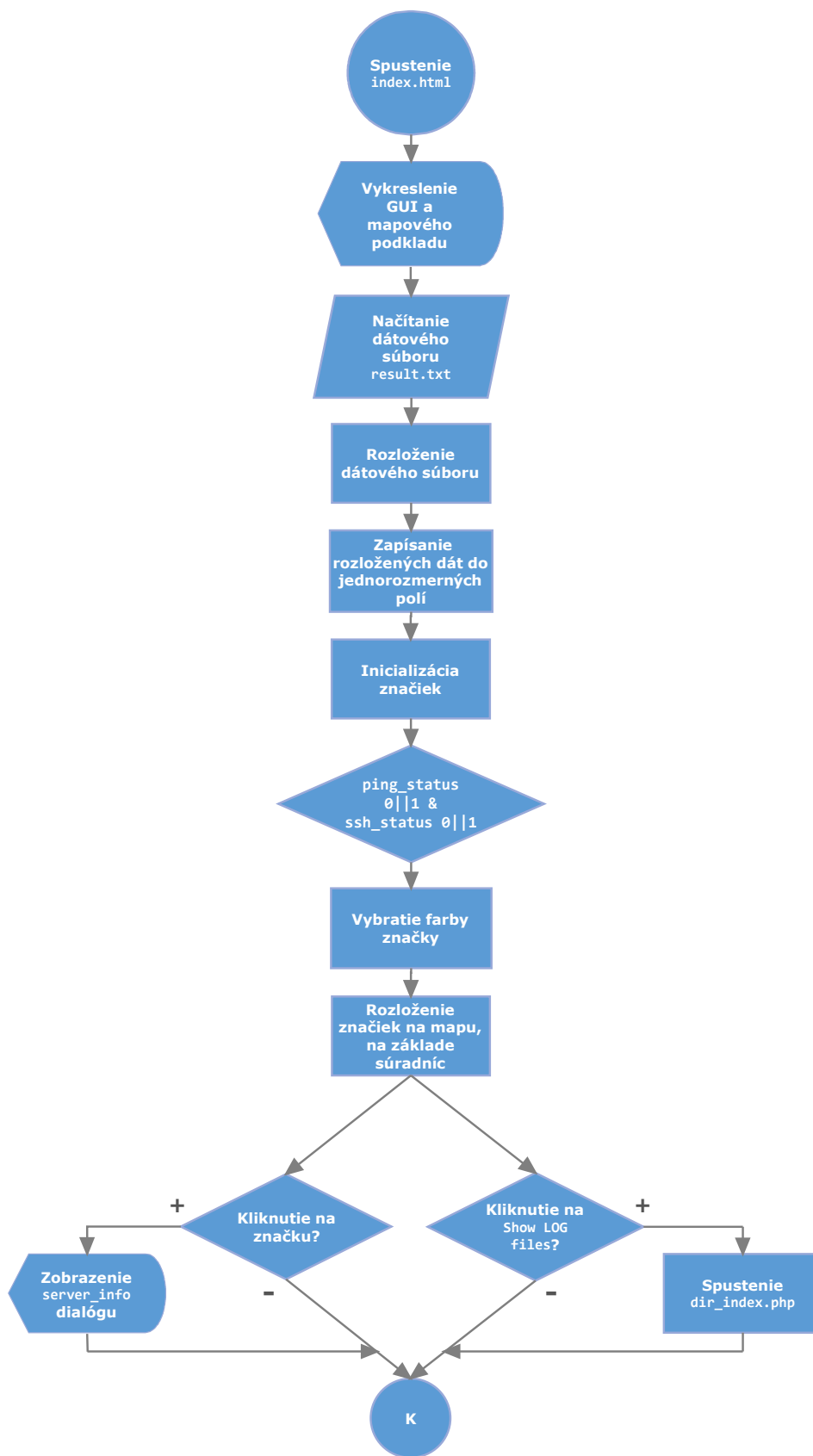
6.5 Nástroj pre zobrazenie súborov LOG

Dodatočné súbory z meraní Ondřeja Pospíchala, ako `metadata.dat`, `pingErr.dat` a ďalšie, sú umiestnené priamo na serveri v podpriechinkoch priečinku LOG. Vzhľadom na to, že dáta sa v danom umiestnení dynamicky menia a dopĺňajú realizoval som zobrazenie týchto dát pomocou PHP skriptu. Pre špecifické nastavenia mojej webovej stránky som upravil skript `file-directory-list`², ktorý je voľne šíriteľný pod licenciou MIT. Po kliknutí na tlačidlo Show LOG files sa spustí spomínaný skript, vygeneruje HTML výstup, vďaka tomu môže užívateľ stránky zobrazovať obsah priečinkov so súbormi LOG, ako aj obsah samotných súborov, ktoré sa otvoria ako textový súbor v okne prehliadača.



Obr. 6.4: Zobrazenie súborov LOG.

²Zdrojový kód je dostupný z <https://halgatewood.com/free/file-directory-list/>.



Obr. 6.5: Vývojový diagram behu webovej aplikácie.

7 ZÁVER

V prvej časti tejto práce som sa venoval popisu výskumnej siete PlanetLab. Popísal som jej stručnú históriu, niekoľko zaujímavých projektov a vysvetlil základnú terminológiu týkajúcu sa tejto siete. V mojej práci sa takisto venujem popisu technológií, ako JavaScript, Google Maps API a webového servera Apache. Cieľom mojej práce bolo vytvoriť interaktívnu webovú stránku, ktorá bude na mapovom podklade zobrazovať uzly siete PlanetLab, na základe dátového súboru, ktorý stránka načíta z dostupného úložiska servera. Tento cieľ mi pomohli splniť práve spomínané technológie. Google Maps API som si vybral hlavne z toho dôvodu, že k nemu existuje na vývojárskych stránkach spoločnosti Google pomerne rozsiahla dokumentácia a takisto veľa užitočných príkladov hotového kódu, pre či už začínajúcich, alebo pokročilých vývojárov. Webový server Apache som volil, zase pre jeho robustnosť, jednoduchosť konfigurácie a bezproblémovosť chodu.

Spojením týchto technológií vznikla fungujúca interaktívna dynamická webová stránka, skladajúca sa z mapového podkladu, značiek zobrazujúcich uzly siete PlanetLab a dynamickej sekcie zobrazujúcej súbory LOG. Každá značka je dynamickým prvkom, ktorý po kliknutí myšou na svoje telo reaguje zobrazením informačného okna s informáciami o danom uzle a textovým oknom obsahujúcim syntax pre manuálne pripojenie pomocou SSH k danému uzlu. Tvar a farba ikony značky je tiež dynamickým prvkom, ktorý sa mení v závislosti na dostupnosti pripojenia SSH a správe typu PING, informácie o ich dostupnosti sú umiestnené v dátovom súbore. Ošetrená bola takisto vhodná pozícia zemepisných súradníc značiek a ich prípadné prekryvanie.

Webová stránka nie je nijako viazaná na aktuálne použitý webový server, je teda veľmi ľahko prenositeľná na akýkoľvek iný webový server a to s len minimálnymi zásahmi do zdrojového kódu. Funkcionalita stránky je takisto veľmi flexibilná, pretože mnou vytvorenú stránku je možné použiť k vykresleniu akýchkoľvek iných bodov na mape, samozrejme za predpokladu použitia vhodného vstupného formátu dát.

Prípadnému pokračovateľovi práce, by som odporučil, pokúsiť sa o vytvorenie živého terminálového okna v prostredí webového prehliadača k priamemu pripojeniu na SSH zvoleného uzlu a to napr. pomocou projektu WebShell. Ďalej je webovú stránku možné rozšíriť o rôzne štatistiky dostupnosti jednotlivých uzlov, prípadne ešte zdokonaľiť grafickú časť tejto webovej stránky.

LITERATÚRA

- [1] PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services. [online], 2007, [cit. 5.5.2016]. Dostupné z URL: <<https://www.planet-lab.org/>>
- [2] J. Navrátil. *PlanetLab - model budoucího Internetu*. Zpravodaj ÚVT MU. ISSN 1212-0901, 2006, roč. XVI, č. 5, s. 1-5.
- [3] RAZVAN BEURAN. *Introduction to network emulation*. Singapore: Pan Stanford, 2013. ISBN 9789814310918.
- [4] M.I.T. Laboratory for Computer Science: Resilient Overlay Networks [online]. [cit. 5.5.2016]. Dostupné z URL: <<http://nms.csail.mit.edu/ron/>>
- [5] ECMA: ECMAScript® 2015 Language Specification [online]. 2015 [cit. 18.5.2016]. Dostupné z URL: <<http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>>
- [6] Steve Suehring. *JavaScript*. 2008, ISBN 9788025122419.
- [7] Dr. Axel Rauschmayer: Categorizing values in JavaScript [online]. Naposledy upravené: 20.1.2013 [cit. 18.5.2016]. Dostupné z URL: <<http://www.2ality.com/2013/01/categorizing-values.html>>
- [8] David Procházka. *PHP 6 začínáme programovat*. 16.01.2012, ISBN 9788024738994.
- [9] Apache Software Foundation: The Apache HTTP Server Project [online]. [cit. 18.5.2016]. Dostupné z URL: <<https://httpd.apache.org/>>
- [10] Netcraft: April 2016 Web Server Survey [online]. 21.4.2016 [cit. 21.4.2016]. Dostupné z URL: <<http://news.netcraft.com/archives/2016/04/21/april-2016-web-server-survey.html>>
- [11] Google: Analýzy a vývoj aplikací [online]. 2016 [cit. 18.5.2016]. Dostupné z URL: <<https://www.google.cz/intx/cs/work/mapsearch/products/mapsapi.html>>
- [12] Google: Google Maps JavaScript API v3. [online]. [cit. 5.5.2016]. Dostupné z URL: <<https://developers.google.com/maps/documentation/javascript/>>
- [13] jQuery: jQuery API [online]. 2016 [cit. 18.5.2016]. Dostupné z URL: <<https://api.jquery.com/>>

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

API	Application Programming Interface
CDN	Content Delivery Network
CSS	Cascading Style Sheets
DNS	Domain Name System
DOM	Document Object Model
GUI	Graphical user interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPD	Hypertext Transfer Protocol Daemon
IP	Internet Protocol
MIT	Massachusetts Institute of Technology
SCP	Secure Copy Protocol
SSH	Secure Shell
URL	Uniform Resource Locator
XML	Extensible Markup Language

ZOZNAM PRÍLOH

A Obsah priloženého CD

37

A OBSAH PRILOŽENÉHO CD

- Elektronická verzia práce.
- Zdrojové kódy webovej aplikácie.
- Skript pre prenos a roztriedenie súborov.