Stran za popravke

|  |  |  |  |
| --- | --- | --- | --- |
| **Stran z napako** | **Vrstica z napako** | **Namesto** | **Naj bo** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Izjava

Podpisani Miha Hren izjavljam, da sem avtor diplomske naloge z naslovom »Spletni vmesnik za izvajanje računskih analiz v visoko-propustnem računskem okolju«.

Izjavljam, da je elektronska različica v vsem enaka tiskani različici.

Izjavljam, da dovoljujem objavo elektronske različice v repozitoriju UL FGG.

Ljubljana, XX. XX. 2012 Miha Hren

Bibliografsko - dokumentacijska stran in izvleček

**UDK:**

**Avtor: Miha Hren**

**Mentor: doc. dr. Matevž Dolenc**

**Naslov: Spletni vmesnik za izvajanje računskih analiz v visoko-propustnem računskem okolju**

**Tip dokumenta: diplomska naloga – univerzitetni študij**

**Obseg in oprema: XX str., XX pregl., XX sl., XX pril.**

**Ključne besede: splet, vmesnik, ajax, html, php, mysql, javascript, condor, git, baza podatkov, izvorna koda**

**Izvleček**

Diplomska naloga je osredotočena na izdelavo spletnega vmesnika za izvajanje računskih analiz v visoko propustnem računskem okolju, ki je vzpostavljeno na FGG in deluje na sistemu Condor.

V prvem delu diplomske naloge so našteta in opisana vsa orodja ter programski jeziki, ki sem jih izbral in uporabljal pri izdelavi spletnega vmesnika. Gre bolj za teoretično poglavje, kjer se bo bralec seznanil z visoko-propustnim računalniškim okoljem ter znanjem, potrebnim za izdelavo spletne strani oziroma spletnega vmesnika. Podani so tudi razlogi za izbiro določenega orodja ali jezika, saj je na voljo še mnogo alternativ.

Drugi del naloge se osredotoči na izdelavo samega spletnega vmesnika. Natančno je razložena vsa izvorna koda, kaj posamezni ukazi pomenijo in seveda namen določenega predela kode. Podani so tudi problemi, ki so se pojavljali tekom izdelave vmesnika, prav tako pa je argumentirana izbira pristopa do reševanja le teh, kjer rešitve niso trivialne.

Najkrajši, tretji del naloge pa opisuje integracijo vmesnika z že vzpostavljenim Condor sistemom na FGG in kakšne težave so se tam pojavile. Opisana so tudi opravila, katera se bodo primarno izvajala preko spletnega vmesnika, ter sistem, s katerim bo izdelani spletni vmesnik komuniciral in ga opravljal.

Bibliographic – Documentalistic information and abstract

**UDC:**

**Author: Miha Hren**

**Supervisor: doc. dr. Matevž Dolenc**

**Title: Web interface for execution of computational tasks in a high throughput computing environment**

**Document type: Graduation Thesis – University studies**

**Notes: XX p., XX tab., XX fig., XX ann.**

**Key words: web, interface, ajax, html, php, mysql, javascript, Condor, git, database, source code**

**Abstract**

This Graduation Thesis is centered around the making of a web interface, to be used for communicating and manipulating with a high-throughput computing system Condor set up at FGG.

The first part of the Thesis contains descriptions of all the tools and programming languages used in the making of the web interface. It is a theoretical chapter, where the reader gets to know high-throughput computing and the knowledge needed to create a web interface. This chapter also contains the reasons for choosing a certain tool or language, as there are many alternatives available.

The second part focuses on the creation of the web interface. Source code gets dissected here, all the specific functions get looked at and explained. Problems and their solutions are also listed here, as well as arguments for said solutions.

The shortes third part describes the implementation of the web interface with an already set up Condor system on FGG. It also contains the information about the different tasks to be executed with Condor, as well as the Condor system itself.

Zahvala

Za pomoč in nasvete pri izdelavi diplomske naloge se iskreno zahvaljujem mentorju doc. dr. Matevž Dolencu.

Posebna zahvala gre tudi mojima staršema in sestri, ki so mi omogočili študij ter mi pomagali z napotki iz studijskih časov ter življenja v splošnem.

Nazadnje bi se rad zahvalil še sošolcem, s katerimi smo preživeli štiri lepa leta, skozi katera smo se z enako zagretostjo zabavali, kot se učili, delali in obiskovali predavanja.

Kazalo vsebine

Kazalo preglednic

Kazalo slik

Kazalo kode

Okrajšave in simboli

PHP angl. PHP: Hypertext Preprocessor

HTML angl. HyperText Markup Language

AJAX angl. Asynchronous Javascript And XML

XML angl. Extensible Markup Language

SQL angl. Structured Query Language

HTC angl. High Throughput Computing

CSS angl. Cascading Style Sheets

HTTP angl. HyperText Transfer Protocol

DOM angl. Document Object Model

GNU angl. GNU's Not Unix

GPL angl. General Public License

OOP angl. Object Oriented Programming

# UVOD

V gradbeništvo je uporaba informacijskih tehnologij vedno večja, pogosto pa tudi nujna. Še najbolj je to opazno v inštitutih in univerzah, kjer stalno potekajo raziskave ter iskanje novih rešitev, za kar pa potrebujejo pomoč prav prej omenjenih informacijskih tehnologij. Ena takih so visoko-propustna računska okolja, ki omogočajo sodelovanje večjega števila naprav pri izračunu nekega kompleksnega problema. Glavna prednost takega sistema je izkoristek obstoječih računalniških naprav, ki se sicer uporabljajo za druge, bolj vsakdanje namene. Posledično pa se znižajo stroški, saj ustanovam ni potrebno kupovati dragih super-računalnikov, katerih namen je izključno reševanje zapletenih problemov v čim krajšem možnem času.

Žal pa uporaba visoko-propustnih računskih okolij ni trivialna. Sistem Condor, kateri je vzpostavljen na Fakulteti za gradbeništvo in geodezijo, se namreč upravlja preko konzolnega okna. Ker je ta neizkušenim uporabnikom neprijazen za uporabo, prav tako pa nepregleden, je smiselno izdelati uporabniški vmesnik. Končni cilj naloge je tako izdelava spletnega vmesnika za izvajanje računskih analiz v visoko-propustnem računskem okolju, ki je vzpostavljeno na FGG in deluje na sistemu Condor. Vmesnik bo ves čas dostopen preko spletnih brskalnikov, saj bo lociran na šolskih strežnikih.

# Teorija

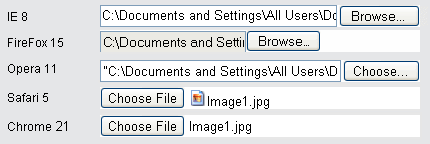
Teoretični del opisuje vsa programska orodja, programske jezike in tehnike, ki so bile uporabljene pri izdelavi te diplomske naloge. Tem kategorijam primerno bo poglavje tudi razdeljeno na podpoglavja. Na kratko bo tudi razloženo, zakaj sem se odločil za uporabo določenega orodja ali jezika napram alternativam.

## Programski jeziki

### HTML

HTML, po angleško HyperText Markup Language, je označevalni jezik, s katerim prikazujemo spletne strani. To počnemo s programi imenovanimi spletni brskalniki, ki pa so pred standardizacijo HTML precej različno interpretirali in prikazovali vsebino HTML gradiv. Ravno zaradi tega je bilo v interesu standardizirati jezik, kar se je z verzijo 2.0 leta 1995 tudi zgodilo. Verzija 1.0 je bila sicer spisana že leta 1990, a je ostala nestandardizirana. Skozi leta je izšlo še nekaj verzij, trenutno aktualna je 4.01, v pripravi pa je že HTML 5.0 standard. Ta bo prinesel kar nekaj potrebnih novosti, ki se nanašajo predvsem na večpredstavnost. Razvoj jezika od leta 1996 dalje vodi organizacija World Wide Web Consortium (<http://www.w3.org/>).

Uporaba HTML jezika je pri izdelavi spletnega vmesnika obvezna, saj gre za standardiziran jezik, ki ga znajo brati vsi današnji spletni brskalniki. Enakovrednih alternativ ni. Potrebno je le paziti na kompatibilnost z vsemi popularnimi brskalniki. Nekatere elemente HTML standarda namreč brskalniki še zmeraj različno interpretirajo. Ena takih je na primer element za nalaganje datotek (Slika 1)



Slika : Prikaz privzetega elementa za nalaganje datotek pri različnih brskalnikih

Sintaksa HTML jezika je med preprostejšimi in preglednejšimi. Gradijo ga namreč HTML elementi, ki so sestavljeni iz značk. Poznamo začetno (npr. <p>) in končno značko (npr. </p>), ki praviloma skupaj tvorita HTML element. Vsebina elementa se nahaja med obema značkama, medtem ko atribute posameznega elementa zapisujemo znotraj prve značke (npr. <p color='black'>vsebina</p>). Spodaj je primer preproste HTML izvorne datoteke (Koda 1). Datoteke s HTML kodo imajo praviloma končnico ''.html'' ali ''.htm''.



Koda : Primer preproste HTML izvorne datoteke

V okviru HTML velja omeniti še CSS, angleško Cascading Style Sheets, ki je z njim neposredno povezan. Gre za poseben način zapisa atributov, povezanih s formatiranjem HTML dokumentov. Sicer jih lahko pišemo tudi znotraj prve značke HTML elementa, vendar zaradi preglednosti raje zapišemo znotraj <head> elementa, ali še bolje, v svojo datoteko. Taka datoteka ima končnico ''.css''.

### PHP

PHP, angleško PHP Hypertext Preprocesor, je skriptni jezik, ki se izvaja na strežniku. Njegov glavni namen je ustvarjanje HTML datotek na podlagi vnesenih vhodnih podatkov. Z drugimi besedami, s pomočjo PHP lahko ustvarjamo dinamične spletne strani. Trenutno je PHP v verziji 5, v prvotni obliki pa je nastal že leta 1995

PHP kodo pišemo v isti dokument kot HTML kodo, pri čemer ju lahko ločimo na več načinov. Najpogostejši in tudi način, katerega sem jaz uporabljal, je pisanje PHP kode znotraj začetnega (<?php) in končnega (?>) ločila. Značilnost jezika je tudi to, da se vse spremenljivke začnejo z dolarskim znakom $. Ostala sintaksa je precej podobna C jezikom, od ustvarjanja in klicanja funkcij, pa do if, while ter for zank. Namesto ''.html'' končnice imajo datoteke s PHP kodo končnico ''.php''. Spodaj (Koda 2) vidimo primer PHP kode s HTML elementom, kjer se vidi razliko med pisanjem komentarjev znotraj obeh jezikov.



Koda : Primer PHP dokumenta s HTML kodo

Pri jezikih, ki se izvajajo na strežnikih, obstaja kar nekaj alternativ, na primer ASP, Java, Perl. Po posvetovanju z mentorjem sem se za PHP odločil z večih razlogov, in sicer:

* Jezik je zelo razširjen po Sloveniji in svetu
* Uporablja se ga lahko brez finančnih stroškov
* Je zelo dobro dokumentiran
* Ni vezan na določen operacijski sistem

Glavna kritika, ki sem jo zasledil v literaturi, je letela predvsem na počasnejšo zmogljivost PHP v primerjavi s konkurenco, kar pa za moje potrebe ni bilo ključnega pomena.

### Javascript

Javascript je skriptni programski jezik, ki je standardno integriran v vseh popularnih spletnih brskalnikih. Izvaja se na strani odjemalca, za razliko od prej omenjenega PHP, ki se izvaja na strežnikih. Njegova glavna značilnost je, da omogoča precej boljšo interaktivnost in uporabniško izkušnjo, prav tako pa je pomemben člen pri programiranju po principu AJAX.

Jezik se je prvič pojavil leta 1995, takrat ga je Netscape ustvaril z namenom poenostaviti programiranje začetnikom. Zaradi tega je imel kar precej kritik, a je vseeno postal standardiziran v okviru razvijanja spletnih strani, s prihodom principa AJAX pa tudi bolj popularen pri profesionalnih razvijalcih.



Koda : Primer javascript kode

Na sintakso je precej vplival C jezik, vendar pa ima javascript nekaj posebnosti. Na primer, spremenljivkam ni potrebno definirati tip, povemo le, da gre za spremenljivko z oznkao ''var'' (glej Koda 3). Prav tako jezik nima klasične podpore objektnega programiranja, pač pa to dosežemo preko posebnih prototipskih funkcij. Za razliko od PHP javascript ne potrebuje posebnih oznak, lahko pišemo znotraj elementa <script> v html kodi.

### SQL

Skoraj vsaka spletna stran potrebuje tudi pripomočke za delo z bazami podatkov in prav za ta namen se uporablja SQL. Po angleško Structured Query Language, je povpraševalni jezik za delo z bazami podatkov. Glavni odliki sta standardizacija in razširjenost, kar pomeni dobro dokumentiranost jezika in lahkost uporabe. Podatki se v bazo zapisujejo v obliki tabel.

Z SQL bazo komuniciramo preko angleških stavkov, ki so zelo podobni naravnemu jeziku. Tako na primer iz neke tabele v bazi poiščemo želen podatek s stavkom ''SELECT polje FROM tabela WHERE pogoji''. Na podoben način ustvarjamo tabele z ukazom ''CREATE'', ali pa zberemo bazo podatkov z ukazom ''USE''.

Jezik je doživel svoj nastanek že v zgodnjih sedemdesetih letih prejšnjega stoletja, bolj pomembna mejnika pa sta leto 1986 in 1987, ko sta ga organizaciji ANSI (American National Standards Institute) in ISO (International Organization for Standardization) standardizirali.



Koda : Primer ustvarjanja tabele v SQL jeziku

## Programska orodja

### Apache

Apache, polno ime Apache HTTP Server, je strežnik za prenos podatkov preko spleta. Deluje na osnovi HTTP-ja (HyperText Transfer Protocol), najbolj razširjenega protokola za komunikacijo med strežnikom in odjemalcem na spletu. Na uradni spletni strani strežnik opisujejo kot robusten, zmožen, primeren za komercialne namene in verjetno še najpomembneje, odprtokoden. Prav tako je strežnik vsakomur dostopen zastonj in dobro dokumentiran, kar je pripomoglo tudi k njegovi popularnosti. Izbira prav tega strežnika kot podlago za testiranje spletnega vmesnika zato ni bila težka.



Slika : Uradni logotip strežnika in fundacije

Projekt Apache je nastal leta 1995, ko je majhna skupina programskih inženirjev skupaj zbrala vsak svoje popravke takratnega predhodnika Apache-ja, z namenom narediti enotno distribucijo strežniškega programa. Ustanovili so The Apache Software Fundation, ki še danes posodablja strežnik. V času tega pisanja je najsodobnejša verzija 2.4.3.

### Condor

Cilj visoko propustnih računskih okolij je združiti večje število računskih virov in tako omogočiti efektivno rabo le teh ter posledično povečati celotno procesorsko moč, ki jo imamo na voljo v danem trenutku. Eden takih okolij je projekt Condor.



Slika : Uradni logotip projekta Condor

Že preko desetletje ga razvijajo na Ameriški univerzi v Wisconsinu, kjer se manj osredotočajo na učinkovitost posameznih naprav in bolj na učinkovitost celotnega sistema naprav, povezanega v omrežje. Najsodobnejša verzija je 7.9.1, na kateri sem tudi testiral uporabniški vmesnik.

Za vodenje, sledenje in upravljanje celotnega sistema skrbi večje število manjših procesov, imenovanih daemons, ki se izvajajo v ozadju na vsaki izmed povezanih naprav. Ti procesi so uporabnikom nevidni in nimajo svojega privzetega vmesnika, vsa komunikacija z njimi poteka preko terminala s pomočjo konzolnih ukazov. To pa me pripelje do glavnega namena te diplomske naloge, in sicer omogočiti bolj laičnim uporabnikom vpogled v različne procese sistema Condor. Slednji je tako jedro, okoli katerega se bo moja diplomska tudi vrtela in način upravljanja s Condorjem bo imel glavni vpliv na obliko in strukturo spletnega vmesnika.

### MySQL

MySQL je skupek orodij za hranjenje, spremljanje in obdelavo baze podatkov. Kot že ime pove, se s strežnikom komunicira s pomočjo SQL jezika. Na voljo je v več paketih, najosnovnejši (MySQL Community Server) je na voljo zastonj. Poudariti je tudi treba, da gre za odprtokodno programsko opremo, zato lahko pri razvoju sodeluje tudi skupnost.



Slika : Uradni MySQL logotip s promocijskim napisom

Okrnjena verzija sicer ponuja zgolj strežnik z dostopom do baze podatkov. Če pa želimo dostop tudi do vseh naprednih orodij za spremljanje in nadzor naše baze, pa moramo poseči po plačljivi Enterprise izdaji. Sicer se z MySQL upravlja preko terminala, seveda z uporabo SQL sintakse, lahko pa se poslužimo tudi zastonjskega MySQL Workbench programa, ki omogoča urejanje in upravljanje baze preko spletnega vmesnika.

MySQL ima trenutno v lasti podjetje Oracle, po tem ko ga je odkupilo od prejšnjega lastnika, Sun Microsystems. V času nastajanja te diplomske naloge je bila aktualna verzija 5.5.28.

### Pluma / Notepad++

Tako Pluma kot Notepad++ sta tekstovna urejevalnika, primarno namenjena pisanju in urejanju izvorne kode. Prvi je pisan za operacijski sistem Linux, drugi pa za Windows okolje.

SLIKA PLUME

Glavni prednosti pred uporabo ostalih tekstovnih urejevalnikov sta sposobnost barvnega označevanja kode različnih programskih jezikov, prav tako pa odpiranje in samodejno ohranjanje zavihkov, ki omogočajo lažje preklapljanje med različnimi datotekami z izvorno kodo. Oba imata pa tudi eno slabost. Ker gre zgolj za urejevalnika teksta, ni omogočeno iskanje napak ter razhroščevanje, kot to omogočajo bolj profesionalni programi za razvijanje, na primer Visual Studio.

SLIKA NOTEPAD++

Oba urejevalnika sta sicer odprtokodna po GNU GPL licenci in sta pod takimi pogoji tudi zastonj dostopna.

### GIT

Git je odprtokodni sistem za nadzor in upravljanje z različicami izvorne kode. Na poseben način omogoča shranjevanje kode in sicer tako, da se shranjuje spremembe glede na zadnjo objavo. Tako imamo vpogled v celotno zgodovino sprememb naše kode in se lahko vrnemo na prejšnje stanje, če trenutna veja razvijanja vodi v prazno. Privzetega uporabniškega vmesnika ni, shranjevanje sprememb se izvaja preko terminalnih ukazov.

SLIKA GIT UKAZOV V TERMINALU

Ukazi za komuniciranje z Git sistemom so preprosti. Začnejo se z besedo ''git'', nadaljujejo pa z želenim ukazom, na primer ''add'' za dodajanje sprememb, ''commit'' za dejansko objavo sprememb v sistem. Seveda se moramo prej v terminalu nahajati znotraj delovne mape, kjer so shranjene tudi vse git datoteke za nek projekt.

SLIKA GITHUB.COM

Sistem je tesno povezan s spletno stranjo github.com, ki omogoča nalaganje projektov, prav tako pa tudi sprememb, na strežnik. To ima dodano vrednost, saj imamo s tem grafični vpogled v različne objave, veje in celotno zgodovino projekta. S pomočjo te spletne strani se tudi precej lažje poveže v skupino in sledi, kdo je kaj prispeval k projektu.

## Tehnike

### AJAX

Asynchronous JavaScript and XML je novejši pristop k programiranju spletnih strani, kjer s pomočjo XMLHttpRequest objekta pošljemo strežniku zahtevo, naj si bo to POST ali GET request, lahko pa tudi kaj bolj zahtevnega, na primer datoteke. Strežnik potem informacije procesira kot ob normalni zahtevi, ter jih preko istega objekta vrne nazaj odjemalcu, kateri jih prikaže. Glavna prednost takega pristopa je predvsem v uporabniški izkušnji. Celotna izmenjava med strežnikom in odjemalcem namreč poteka asinhrono v ozadju. Uporabnik tako ni izpostavljen ponovnemu nalaganju celotne strani, ampak se posodobi le določen del le te.

Pri izvedbi AJAX principa se uporablja več tehnologij, predvsem HTML in CSS za prikaz in izgled, DOM in XML za dinamični prikaz in interakcijo s podatki, ter seveda javascript, ki vse tehnologije poveže in tudi vsebuje XMLHttpRequest objekt. Tako gre pri AJAX-u za skupek tehnologij, ki nam omogočajo bolj moderno in uporabniku prijazno prikazovanje spletnih strani.

### Objektno orientirano programiranje

Pri objektno orientiranem programiranju (krajše OOP) gre za paradigmo reševanja problemov, kjer kodo strukturiramo v objekte, s tem pa občutno zmanjšamo količino proceduralne kode in jo naredimo bolj pregledno, lažje berljivo.

Razred je osnovni koncept objektnega programiranja. Gre za neke vrste načrt oziroma strukturo objekta, kjer so definirane vse spremenljivke (podatki in atributi) in vse funkcije (metode), ki jih lahko nek objekt uporablja. S pomočjo teh funkcij in spremenljivk lahko potem manipuliramo z instanco razreda, ki ji rečemo objekt.

OOP koda je tako zgrajena iz številnih objektov, ki se med sabo prepletajo in tvorijo nek smiseln algoritem. Glavne prednosti takega načina programiranja so:

* Bolj pregledna koda kateri je lažje slediti
* Hitrejše iskanje in popravljanje napak
* Omogoča lažje in bolj učinkovito delo v skupini
* Koda je lažje prenosljiva tudi na druge projekte

Glavna slabost pa je predvsem ta, da je na konceptualni ravni tako kodo težje dobro strukturirati in oblikovati.

# Uporabniški vmesnik

PRILOGE

Priloga A Vsebina datoteke database.data

Priloga B Lastnosti, ki jih lahko določimo posamezni šobi

Priloga C Pregled menijev programa Smokeview

C1 Load/Unload

C2 Show/Hide

C3 Options

C4 Tour

C5 Reset

C6 Help

C7 Quit

Viri

Miha Hren