

Inteligentný monitorovací systém práce študentov v Linuxovom termináli

Adrián Matejov
Školiteľ: RNDr. Marek Nagy, PhD.

Fakulta matematiky, fyziky a informatiky
Univerzita Komenského v Bratislave

27. jún 2018

- Zadania úloh v moodli
- Žiadna interaktívnosť
- Ťažké overovanie správnosti riešenia
- Žiadny prehľad pre vyučujúceho o riešeniach

Linux pre používateľov 2016/2017

Z príkazového riadka sa presunte do priečinka `/tmp`.

```
[cvicenie_02 l00][adman@codemachine:/tmp/selfgz967631143]$ cd /tmp
```

Teraz sa nachádzate v priečinku `/tmp`, tento fakt si môžete overiť aj príkazom `pwd`. Vytvorte v priečinku `/tmp` podpriečinok s názvom `uloha02`.

```
[cvicenie_02 l01a][adman@codemachine:/tmp]$ mkdir uloha02
```

Teraz prejdite do vytvoreného priečinka `uloha02`.

```
[cvicenie_02 l02a][adman@codemachine:/tmp]$ cd uloha02/
```

Vo Vami novo vytvorenom priečinku sme Vám vyrobili súbor `blackbox.pl`. Tento súbor je malý perl script, ktorý Vám vytvorí súborovú/priečinkovú štruktúru v momentálnom priečinku. Tento súbor ale najskôr treba spustiť. Keby sme napísali teraz do príkazového riadku len názov súboru, príkaz nezafunguje, pretože Linux ešte nevie, akého typu je tento súbor. Preto mu to najskôr treba povedať. Pre vytvorenie súborovej/priečinkovej štruktúry teda spustíte príkaz

```
$ perl blackbox.pl
```

```
[adman@codemachine:/tmp/uloha02]$ perl blackbox.pl
```

Obr.: Ukážka GTA cvičenia

go-term-adventures (GTA)

- Interaktívne cvičenia
- Zadania úloh priamo v termináli
- Automatické testovanie riešenia

Problémy

- Veľa študentov (cca 90) a málo cvičiacich
- Žiadny prehľad o pokroku študentov v reálnom čase
- Ťažké zisťovať, kto má problémy s úlohou
- Nemožnosť automaticky obodovať riešenia
- Nemožnosť analýzy rôznych spôsobov riešenia úloh

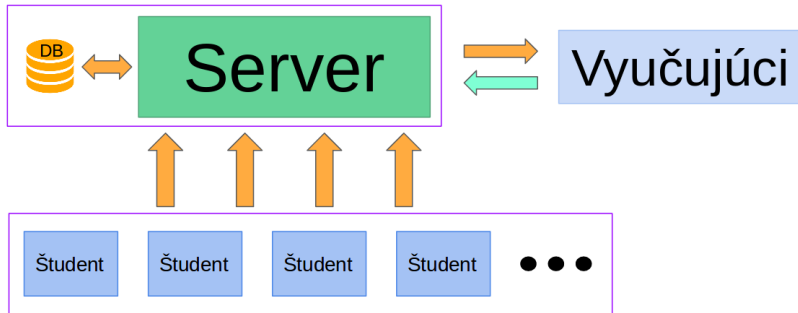
GTA-Monitor - monitorovanie priebehu cvičenia

- Prijímanie dát z GTA aplikácie (začiatok cvičenia, odoslanie príkazu, vyriešenie úlohy, ukončenie cvičenia...)
- Ukladanie dát do databázy
- Real-time aktualizovanie klientov cez websocket
- Vizualizácia v prehliadači
- Detekcia neaktívnych/neúspešných študentov

GTA-Monitor - po cvičení

- Automatické hodnotenie cvičenia
- Základné štatistiky
- Klasifikácia rôznych spôsobov riešenia - k-means

GTA-Monitor - monitorovanie priebehu cvičenia



Obr.: Architektúra toku dát

Technické detaily

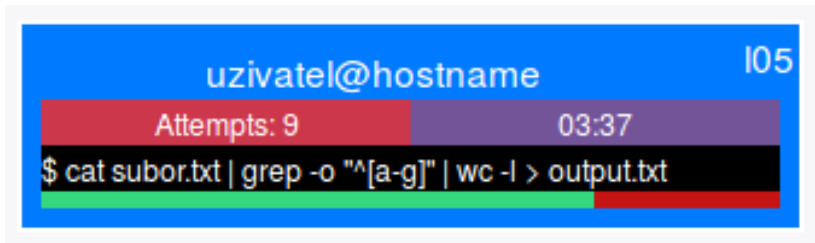
- backend - nodejs + express framework
- komunikácia cez websockets - socket.io
- databáza - mysql (+ sequelize ako framework)
- frontend - html/js/css
- wget - odosielanie dát od študentov do systému

GTA-Monitor - monitorovanie priebehu cvičenia



Obr.: Vizualizácia miestnosti

GTA-Monitor - monitorovanie priebehu cvičenia



Obr.: Vizualizácia pokroku študenta

Klasifikácia rôznych spôsobov riešenia

Každé riešenie reprezentované zoznamom slov

```
["cat", "file.txt", {op: "|"}, "grep", "^ab[0-9]+",  
{op: "|"}, "grep", "-n", "-o", "[0-9]+",  
{op: ">"}, "output.txt"]
```

Unigramy

```
{cat, file.txt, |, grep, ^ab[0-9]+, -n, -o, [0-9]+, >,  
output.txt}
```

Špeciálne bigramy

```
{cat, cat file.txt, |, grep, grep ^ab[0-9]+,  
grep -n, grep -o, grep [0-9]+, >, output.txt}
```

Klasifikácia rôznych spôsobov riešenia

- Množina pretransformovaná na 'bag of words' vektor s hodnotami 0/1
- *K-means*
 - Jaccardova vzdialenosť

$$1 - \frac{|A \cap B|}{|A \cup B|}$$

- kosínusová vzdialenosť

$$1 - \frac{A \cdot B}{\|A\| \|B\|}$$

Klasifikácia rôznych spôsobov riešenia

Spätná väzba pre učiteľa (k-means zhluky)

- `cat histogram.txt | sort -t" " -k 2 -n > counts.txt`
- `sort histogram.txt -n -k2 > counts.txt`
- `tac histogram.txt | sort -n -k2 > counts.txt`

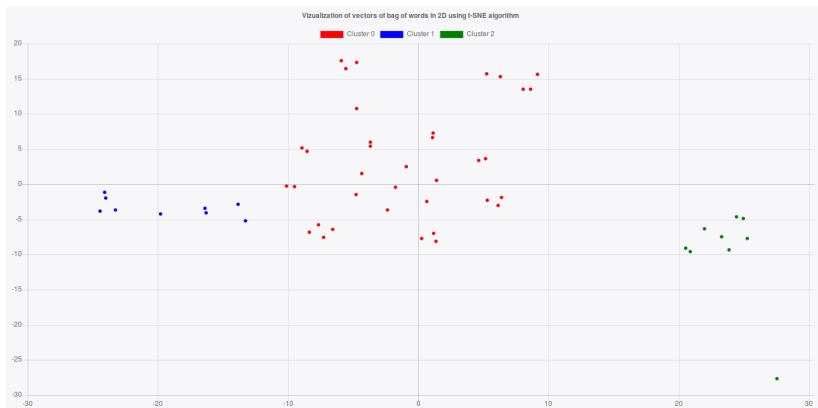
Klasifikácia rôznych spôsobov riešenia

- 1 `grep "[^:]*$" -o /etc/passwd > zoznam.txt`
- 2 `grep "[^:]*$" -o /etc/passwd > ../zoznam.txt`
- 3 `cat /etc/passwd | grep -o "[^:]*$" > zoznam.txt`

	Jaccardova	Kosínusová
Unigramy (1), (2)	0.28571	0.16666
Unigramy (1), (3)	0.25	0.13397
Unigramy (2), (3)	0.44444	0.27831
Špec. bigramy (1), (2)	0.28571	0.16666
Špec. bigramy (1), (3)	0.5	0.31958
Špec. bigramy (2), (3)	0.63636	0.45566

Tabuľka: Porovnanie vzdialeností reprezentácií vstupov (1), (2), (3)

Klasifikácia rôznych spôsobov riešenia



Obr.: Vizualizácia riešení v 2D priestore pomocou t-SNE algoritmu

Testovanie aplikácie

- 10 cvičení (február 2018 - máj 2018)
- 7 cvičiacich - pozitívne hodnotenia aplikáciu
- 56 študentov

Na základe zozbieraných historických dát

- Vedieť odhadnúť, koľko by daná úloha mala približne trvať
- Lepšia heuristika na detekciu zaseknutých študentov
- Zaradiť študentov do istých „kategórií problémovosti“

Integrácia systému na iných kurzoch

- Aplikovateľný aj na iné dómeny výuky - Programovanie v pythone, Scratch, C++...
- Pridanie odosielania výsledkov do testovačov - API je jednoduché
- Rozšírenie tohto systému pre podporu rôznych kurzov

Ďakujem za pozornosť

<https://github.com/NaiveNeuron/gta-monitor>

Otázky a pripomienky

“Prečo v tabuľke „Hall“ máte primárny kľúč, keď sa nikde nepoužíva (neodkazuje sa naň)?”

- Momentálne ho netreba
- V budúcnosti rozšírenie aplikácie pre zobrazovanie rôznych miestností = viac záznamov pre jedného užívateľa

Otázky a pripomienky

```
"hostname": "'hostname'"  
"Prečo je druhý hostname v apostrofoch?"
```

Apostrofy hovoria, že príkaz `hostname` bude interpretovaný shellom a výstup sa vráti.

Otázky a pripomienky

“Čo znamená hash úspešnosti cvičenia? Na čo sa používa?”

"i" + *názov cvičenia* + "j%!d(string=" + *názov úlohy* + ")k" +
domovský priečinok + "l"

- Použitie pri pridelovaní bodov za cvičenie
- Prevencia pri pokuse o „hacknutie“

Otázky a pripomienky

“Čo sa stane, ak má nejaký vektor presne rovnakú vzdialenosť od dvoch a viac centroidov?”

```
for(let i=0, max = v.length; i < max; ++i) {
  let min = groups[0].distances[i];
  let indexGroup = 0;
  for(let j=1, max2=groups.length; j < max2; ++j){
    if (groups[j].distances[i] < min) {
      min = groups[j].distances[i];
      indexGroup = j;
    }
  }
}
```

Listing 1: Kód z knižnice node-kmeans