

Iskanje podobnih video posnetkov kot spletna storitev v oblaku

Smiljan Šinjur, Jurij Munda

UM FERi, Smetanova 17, 2000 Maribor, Slovenija

E-pošta: smiljan.sinjur@um.si

Video matching as a web service in cloud computing

A video matching algorithm was developed and runs in our own implementation of the SaaS cloud. The cloud's nodes support web services, computations, storage capacities and networking. The video matching algorithm is distributed among computations nodes that utilize only central processing units. Obtained performance is very promising and can also be improved by deploying graphical processing units.

1 Uvod

V članku opisujemo spletno storitev, ki išče dvojnike video posnetkov. Iskalni algoritem se izvaja v računalniškem oblaku, saj se s tem izognemo potrebi po nakupu nove strojne opreme ter tudi zmanjšamo čas, namenjen vzdrževanju. Skupaj z drugimi storitvami tudi enakomerneje in optimalneje izkoristimo oblačne komponente.

Računalništvo v oblaku pomeni pristop, pri katerem so dinamično razširljivi računalniški viri na voljo kot storitev v računalniškem omrežju [1]. Najbolj prepoznavni ponudniki računalništva v oblaku so Amazon Web Services, Google, Microsoft in Rackspace, iz Slovenije pa omenimo podjetje Virtu (Flip IT) in nekomercialno organizacijo ARNES. Od leta 2011 vse več ponudnikov oblačnega računalništva uporablja ad-hoc standard OpenStack [2], ki omogoča povezovanje komponent različnih ponudnikov računalništva v oblaku.

Pri računalniški obdelavi slik potrebujemo zmogljive računalniške arhitekture, saj so obdelovalni algoritmi lahko zelo zahtevni. Takšen je problem tvorbe značilnic, ki uporablja statistične metode, na primer tvorbo hierarhičnih značilnic iz barvnega histograma [3,4], ali pa algoritme z razliko Gaussovih jeder. Sem spadata tudi zelo znan postopek SIFT [5] in njegova nadgradnja PCA-SIFT [6].

Iskanje ujemanja različnih značilnic je prav tako dobro raziskan in računsko zahteven problem. Avtorji [7] pri iskanju uporabijo lokalno občutljivo sekljalno tabelo (LSH - *Local Sensitive Hashing*). Obstajajo tudi aproksimativni iskalni algoritmi NN (*Nearest Neighbour Search*) [8].

Članek je razdeljen v 6 poglavij. V drugem poglavju opišemo zasnovano spletno storitev. Temu poglavju sledi opis arhitekturne, uporabljene za izvajanje nalog v oblaku. Zasnovano algoritma za tvorbo video značilnic in iskanje podobnosti med njimi razložimo v četrtem

poglavju, v petem algoritem in rezultate analiziramo, s šestim pa članek sklenemo.

2 Spletna storitev

V Laboratoriju za sistemsko programsko opremo smo razvili sistem, ki omogoča zunanjim partnerjem enostaven dostop do naših storitev. Odločili smo se, da že obstoječo laboratorijsko infrastrukturo prilagodimo novim potrebam. Glavni izziv je bil zagotoviti ustrezno računsko moč, da nam ne bi bilo treba spreminjati obstoječih algoritmov, ki niso bili razviti za delo v oblaku.

Komponente razvitega sistema so: spletna aplikacija, baza uporabnikov, diskovni prostor za uporabniške zbirke, razvrščevalnik opravil in množica računskih vozlišč, ki izvajajo algoritme in rezultate shranjujejo v podatkovno bazo ali na uporabniški diskovni prostor.

Spletna aplikacija omogoča prijavo uporabnika v sistem, nalaganje podatkov, ki jih želi obdelati z našimi algoritmi, zagon izbranih algoritmov in pregled rezultatov. Uporabniku lahko dodelimo tudi diskovne kapacitete, na katere lahko shrani večje zbirke, ki jih želi poslati v obdelavo.

Zahtevo za izvajanje določenega algoritma z uporabnikovimi podatki imenujmo »posel«. Ko uporabnik pošle posel, se ta postavi v čakalno vrsto. Posli se izvajajo v računskih vozliščih. Vsako vozlišče lahko sočasno izvaja zgolj en posel, delo pa jim dodeljuje razvrščevalnik. Vozlišča imajo vnaprej določeno prioriteto (glede na svoje zmogljivosti), razvrščevalnik pa za vsak nov posel poišče pristo vozlišče, pri čemer igra prioriteta vozlišč glavno vlogo. Razvrščevalnik lahko upravlja poljubno število vozlišč. Na sistemu obstaja več vrst poslov (uporablja se tudi za pedagoške namene z različnimi prioritetami. Najvišjo prioriteto imajo posli, ki jih ustvarijo administratorji. Sledijo posli, ki jih ustvarjajo storitve, nato so posli, ki jih ustvarijo študenti, na zadnjem mestu pa so posli, ki jih ustvari sistem za pomoč pri ocenjevanju študentskih nalog. Razvrščevalnik zato uporablja kombinacijo dveh strategij razvrščanja: prioriteto med različnimi tipi poslov in FIFO (*first in first out*) med posli istega tipa. Razvrščanje poteka neprekinjevalno – šele ko se posel na računskem vozlišču konča, mu lahko razvrščevalnik dodeli nov posel.

Računsko vozlišče predstavlja navidezni ali fizični računalnik z nameščenim operacijskim sistemom linux. Trenutno uporabljamo 64-bitno distribucijo Ubuntu 14.04 Server. Ugotovili smo, da mora za naše posle računsko vozlišče izpolnjevati naslednje minimalne

zahteve: vsaj 1 procesorsko jedro (raje 2-4) in vsaj 3 GB pomnilnika. Za izvedbo posla potrebuje računsko vozlišče še nekaj podatkov. Številka posla ga poveže s konfiguracijsko zbirko, ki jo je uporabnik oddal v obdelavo; številka algoritma mu pove, kateri algoritem naj požene; številka uporabnika pa mu omogoča, da uporabnika identificira ter dostopa do podatkov, ki jih je uporabnik shranil na svoj diskovni prostor v našem oblaku. Rezultati izvedenega posla so lahko poročilo (vse, kar algoritem izpisuje na standardni izhod, se shrani v poročilo), krajše zbirke (npr. slike) ter velike zbirke, ki pa se ne shranijo v podatkovno bazo, temveč se shranijo na uporabnikov diskovni prostor v oblaku.

3 Oblak

Uporabljeni oblak je plod razvoja internega informacijskega sistema, ki poteka že od leta 1997 [9]. Glede na definicijo oblaka po NIST [10] gre za zasebni oblak, ki sodi v kategorijo SaaS (*Software as a Service*).

V oblaku trenutno teče več storitev: sistem za delo s študenti (distribucija pedagoških vsebin in oddajanje nalog), storitev za samodejno preverjanje nalog (študenti lahko pred oddajo nalogo pošljejo v »avtomatsko preverjanje«, kjer se njihova naloga samodejno ovrednoti in o tem dobijo poročilo), sistem za pomoč pri ocenjevanju oddanih nalog (v bistvu gre za poglobljeno preverjanje študentskih izdelkov, za katere lahko sistem predlaga tudi oceno), sistem za iskanje programskih plagiatov [11], sistem za generiranje in ovrednotenje pisnih (papirnatih) preizkusov z vprašanji zaprtega tipa [12] (podpira tudi odprta vprašanja, a slednja mora pedagoški delavec ovrednotiti sam) in sistem za poganjanje storitev, namenjenih zunanjim partnerjem.

Oblak sestavljajo štiri glavne komponente: računalniški sistemi, ki nudijo storitve, računalniški sistemi, ki nudijo računsko moč storitvam, računalniški sistemi, ki nudijo diskovne kapacitete, in omrežje.

V oblaku imamo 3 omrežja: internet (zunanj IP-naslovi), intranet (notranje omrežje, ločeno od zunanjega sveta) in omrežje SAN (*storage area network*). Vsak izmed računalniških sistemov je lahko priključen na eno ali več omrežij – odvisno od zahtev.

SAN-omrežje nudi računalniškim sistemom v oblaku ustrezne diskovne kapacitete. Običajno jih nudi po protokolih iSCSI (*Internet Small Computer System Interface*) ali CIFS (*Common Internet File System*).

Računalniški sistemi, ki nudijo storitve, so virtualizirani. Na ta način smo zelo poenostavili vzdrževanje in nadgrajevanje sistemov (npr. povečanje pomnilnika, števila jeder in diskovnih kapacitet).

Računalniški sistemi, ki nudijo računsko moč (računska vozlišča) so lahko virtualizirani ali pa fizični. Za slednje se odločimo, ko začasno potrebujemo večjo računsko moč ali pa imamo storitve s posebnimi zahtevami (npr. uporabo grafične kartice za izračun). V osnovi je programska oprema na vseh računskih vozliščih identična, tako da lahko poganjajo vse

osnovne algoritme, se pa lahko razlikujejo po zmogljivosti.

Strojna oprema, na kateri teče naš oblak, ima trenutno dve glavni enoti: strežnik IBM x3650M4 in strežnik Synology RS2414RP+. Na x3650M4 je nameščena brezplačna različica virtualizacijskega sistema vmWare ESXi 5.5. Ta strežnik ima tudi znatno količino notranjih diskovnih kapacitet (6 SAS-diskov 1 TB povezanih v polje RAID 5). Interni diski omogočajo, da sistemi preživijo izpade omrežja na fakulteti (v zadnjih dveh letih smo imeli na fakulteti dva večja izpada omrežja, ker so se CISCO-va mrežna stikala kaskadno zrušila). Za diskovne kapacitete skrbi strežnik Synology NAS (*Network Attached Storage*), na katerem teče programska oprema DSM 4.3. Za dodeljevanje diskovnih kapacitet uporabljamo skopo oskrbovanje (*thin provisioning*), kar pomeni, da lahko dodelimo več diskovnega prostora, kot ga je na voljo. V sistemu imamo trenutno 5 SATA-diskov 4 TB, povezanih v polje RAID 5. Sistem ima na voljo še 10 mest za diske, možno pa je dokupiti tudi razširitveno enoto za dodatnih 15 diskov.

Zaradi varnosti in izkušenj iz preteklosti je celoten oblak podvojen, in sicer s strojno opremo prejšnje generacije. Rezervni oblak predstavlja strežnik IBM x3400 s 6 SATA-diski 1 TB, povezanimi v polje RAID 5, in NAS-strežnik IBM FASt600 s 24 FC-diski 73 GB, povezanimi v več polj RAID 5. Slednji sistem je bil donacija podjetja NovaKBM, d.d., ko so ga zaradi poteka vzdrževalne pogodbe odstranili iz produkcije. Kljub že kar častitljivi starosti (11 let) po hitrosti še kar konkurira današnjim NAS-sistemom (srednjega ranga), saj je s strežnikom povezan s FC-povezavo 2 Gbps, diski pa se vrtijo s 15.000 rpm. Računalniški sistemi v glavnem in rezervnem oblaku se sinhronizirajo dvakrat dnevno. Oblaka bi lahko sicer tudi združili, a za sedaj nam takšna strogo ločena delitev ustreza.

Ko smo pred leti naredili prve korake na poti, ki nas je pripeljala do sedanje točke, smo imeli kar nekaj pomislekov, ali je uporaba enega močnega in razmeroma dragega strežnika smiselna, saj smo sprva uporabljali več osebnih računalnikov. Po prehodu na strežniški razred računalniške strojne opreme in uvedbi virtualizacije (sprva smo uporabljali Xen-virtualizacijo), se je izkazalo, da je ta prehod navkljub večjemu začetnemu strošku upravičen. Čas, ki ga danes uporabimo za vzdrževanje in nadgrajevanje strojne opreme, je postal skoraj zanemarljiv. Tudi zanesljivost strojne opreme se je s prehodom na višji razred strojne opreme zelo izboljšala. Pri osebnih računalnikih je bil čas do odpovedi prve komponente strojne opreme približno leto dni (običajno trdi disk ali napajalnik), pri našem prvem »pravem« strežniku IBM x3400 pa je do odpovedi prvega kosa strojne opreme (interni SATA-disk) minilo kar 37 mesecev.

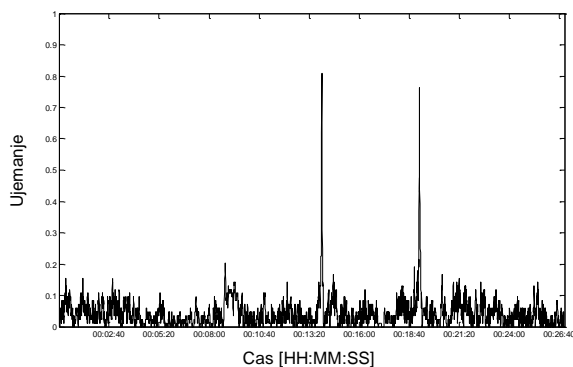
Pri našem oblaku je poskrbljeno tudi za primer izpada električne energije. Vsa strojna oprema (razen mrežnih stikal) ima redundantno napajanje. En napajalnik je priključen na mrežno napajanje

neposredno, drugi pa je povezan s sistemom za brezprekinitveno napajanje (UPS). UPS-i niso priključeni na javno omrežje, temveč so na omrežje fakultetnega električnega agregata.

4 Tvorba slikovnih značilnic in iskanje videoposnetkov

Ena izmed storitev, ki teče v našem oblaku, je iskanje ujemaajočih se video posnetkov. Video posnetke opredelimo kot podobne, če se razlikujejo v osvetlitvi, manjših lokalnih deformacijah, odstopanjih zaradi različnih kodekov, resoluciji ali celo zaradi delne izgube vsebine.

Uporabnik v prvem koraku v oblak prenese video posnetke, ki jih želi obdelovati. V drugem koraku iz video posnetkov tvorimo značilnice, ki jih v tretjem koraku uporabimo za določanje ujemanja med video posnetki. Rezultati ujemanja so prikazani grafično, tako da za vsak testni par video posnetkov rišemo stopnjo ujemanja (os y) v odvisnosti od trajanja daljšega posnetka (os x) – slika 1.



Slika 1: Primer prikaza za ujemanje dveh video posnetkov

Slika kaže, da se krajši video posnetka v trajanju 14 s ujema s segmentoma, ki se v daljšem posnetku začneta pri 13. minuti in 30. sekundi ter pri 18. minuti in 50. sekundi.

4.1 Tvorba značilnic

Kot osnovo za tvorbo značilnic vzamemo sliko (*frame*) iz video posnetka. Najprej jo pretvorimo v sivinsko sliko. Vsako sivinsko sliko nato večnivojsko pragovno segmentiramo [13]. Dodatno pohitritev tovrstne segmentacije dosežemo z uporabo binarnih operacij, tako da sliko razdelimo v največ 2^n slikovnih ravnin, pri čemer je n barvna globina, torej $n = 1, \dots, 8$. Vsaka segmentna slika je binarna, ker velja pravilo: če se sivina izbranega piksla nahaja v območju i -te slikovne ravnine, bo piksel na pripadajoči izhodni sliki črn, drugače pa bel.

Iz črnih pikselov vsake segmentne slike tvorimo vgnezdene izbočene lupine [14]. V splošnem je časovna zahtevnost tvorbe vgnezdenih izbočenih lupin $O(n \log(n))$. Vendar pa točke v našem primeru postavimo na poravnano mrežo, tako da dosežemo linearno časovno zahtevnost [15]. Dodatna ključna prednost takšne tvorbe lupin možnost paralelizacije, če seveda strojna oprema to dopušča.

Vgnezdene izbočene lupine opisujejo vsebino slike, iz njih pa izluščimo značilnico \mathbf{x} , ki je vektor vrednosti. Element i v vektorju pomeni število vgnezdenih lupin v i -tem segmentu oz. na i -ti slikovni ravnini:

$$\mathbf{x} = [d^{(1)}, d^{(2)}, \dots, d^{(i)}, \dots, d^{(n)}]. \quad (1)$$

Značilnico video posnetka sestavljajo značilnice njegovih slik.

4.2 Iskanje ujemanja

Pri iskanju ujemanj med video posnetki nas prvenstveno zanima, na katerem mestu se krajši posnetek pojavi v daljšem in s kakšno stopnjo podobnosti.

Podobnost dveh slik lahko definiramo kot evklidsko razdaljo njunih značilnic \mathbf{x} in \mathbf{y} :

$$l = \|\mathbf{x} - \mathbf{y}\|. \quad (2)$$

Če sta vektorja značilnic normirana, bo njuna razdalja vedno med 0 in 1, hkrati pa bo veljalo, sta si sliki tem bolj podobni, čim manjša je ocenjena razdalja.

Nadalje lahko določimo stopnjo podobnosti med dvema video posnetkoma:

$$L_{i,j} = \frac{1}{k} \sum \left\| l_{i-\frac{k}{2}, i+\frac{k}{2}} - l_{j-\frac{k}{2}, j+\frac{k}{2}} \right\|, \quad (3)$$

pri čemer je i indeks slike iz prvega videa, j indeks slike iz drugega videa, k pa določa okoliške slike iz obeh videov, ki so vključene v primerjavo.

V večini primerov iščemo krajši video posnetek znotraj daljšega, zato lahko indeksa j in k za krajši video fiksiramo na polovico posnetkove dolžine.

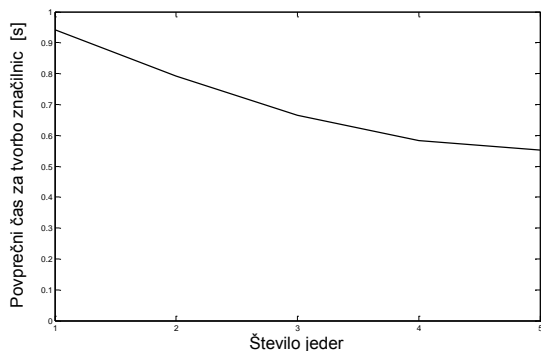
5 Analiza in rezultati

Pri izvedbi algoritma za iskanje dvojnikov smo želeli minimizirati čase za tvorbo značilnic in za poizvedbe. Ugotovili smo tudi, pri katerih parametrih je iskanje lahko realnočasovno, pogreški pri ujemanjih pa so minimalni.

Algoritem za tvorbo značilnic ima več ključnih prednosti: linearno časovno zahtevnost, paralelno tvorbo značilnic in možnost, da video posnetke ob tvorbi značilnic podvzorčimo.

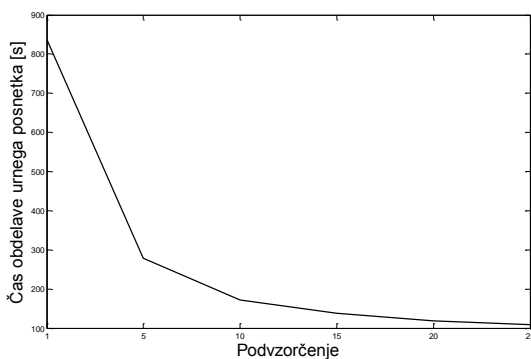
Preizkuse smo izvedli z entiteto opisanega oblaka, in sicer na računalniku z osrednjo procesno enoto Intel i5 650 z dvema jedroma in štirimi nitmi ter z 8 GB delovnega pomnilnika. Resolucija vseh video posnetkov je 640×480 .

Slika 2 prikazuje povprečen čas, potreben za tvorbo značilnic ob spreminjanju števila procesnih jeder. Slike smo razgrajevali na 8 ravnin (segmentov). Čas tvorbe značilnic ob uporabi samo enega jedra je v povprečju 0,9 s, ob uporabi 4 jeder pa se čas tvorbe značilnic zmanjša na 0,6 s.



Slika 2: Povprečni čas za tvorbo značilnic, če uporabimo različno število procesnih jeder

Slika 3 prikazuje čas, potreben za tvorba značilnic video posnetka, kadar ne upoštevamo vseh zaporednih slik, ampak le vsako L -to (podvzorčenje). Graf prikazuje povprečen čas, potreben za tvorbe značilnic pri enournem video posnetku. S podvzorčenjem 25 (upoštevana le vsaka 25. slika) znaša 109 s.



Slika 3: Čas tvorbe značilnic za enurni video posnetek pri različnih stopnjah podvzorčenja

6 Razprava in sklepi

Spletna storitev za iskanje podobnih video posnetkov, ki se izvaja v oblaku, se je izkazala kot primerna za demonstracijske namene, za komercialno rabo pa bi potrebovala bolj dovršen uporabniški vmesnik.

Ugotavljamo, da računalniški oblak razvijamo v pravo smer, saj se lepo ujema komponentami, ki jih predvideva standard OpenStack. Lastne oblačne rešitve in programska oprema imajo utečen in razmeroma kratek cikel razvoja in preizkušanja, kar omogoča razmeroma hitro prilagajanje novim zahtevam.

Delovanje oblačne storitve za iskanje podobnih videov je zelo obetajoče, saj smo uspešno izkoristili večprocesno tvorbo vgnezenih izbočenih lupin. Pokazalo se je, da lahko ob le manjši izgubi natančnosti pri določanju ujemanja izvajalne čase bistveno skrajšamo, če ne tvorimo lupin za čisto vse slike video posnetka.

Čeprav že sedaj s primerno izbiro vhodnih parametrov algoritmi tečejo zelo hitro, pričakujemo, da bi izvajanje lahko še občutno pohitrili, če bi del

programske kode selili na grafične procesorje. S tem bi dosegli, da bi bili tudi pri daljših posnetkih in natančnejših primerjavah odzivni časi le nekaj minut.

Literatura

- [1] http://en.wikipedia.org/Cloud_computing, 21. 7. 2014.
- [2] http://en.wikipedia.org/Open_Stack, 21. 7. 2014.
- [3] O. Chum, J. Philbin, M. Isard in A. Zisserman: »Scalable near identical image and shot detection,« Proceedings of the 6th ACM international conference on Image and video retrieval, CIVR '07, str. 549-556, New York, NY, ZDA, 2007.
- [4] J.-M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders in H. Geerts: »Color invariance,« IEEE Trans. Pattern Anal. Mach. Intell., vol. 23 str. 1338-1350, 2001.
- [5] J. Sivic in A. Zisserman: »Video google: A text retrieval approach to object matching in videos,« Proceedings of the Ninth IEEE International Conference on Computer Vision, vol. 2, str. 1470-1477, Washington, DC, ZDA, 2003.
- [6] D.G. Lowe: »Object recognition from local scale-invariant features,« The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, str. 1150-1157, 1999.
- [7] Y. Ke in R. Sukthankar: »Pca-sift: A more distinctive representation for local image descriptors,« CVPR (2), str. 506-513, 2004.
- [8] K. Grauman in T. Darrell, »Efficient image matching with distributions of local invariant features,« Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, str. 627-634, Washington, DC, USA, 2005. IEEE Computer Society.
- [9] M. Lenič idr.: »Information system for laboratory work management,« 5th Euromedia conference 2000, Antwerp, Belgija, 8.-10. maj 2000, str. 245-249.
- [10] P. Mell, T. Grance: »The NIST definition of Cloud Computing,« Special Publication 800-145, National Institute of Standards and technology, U.S. Department of Commerce, september 2011, 7 strani.
- [11] J. Munda, M. Lenič: »Zaznavanje plagiatov v programskem inženirstvu,« Zbornik štirinajste mednarodne Elektrotehniške in računalniške konference ERK 2005, 26. - 28. september 2005, Portorož, Slovenija, zv. B, str. 356-359.
- [12] J. Munda: »Avtomatsko generiranje in ocenjevanje preverjanj z vprašanji zaprtega tipa,« Zbornik sedemnajste mednarodne Elektrotehniške in računalniške konference ERK 2008, 29. september - 1. oktober 2008, Portorož, Slovenija, zv. B, str. 321-324.
- [13] M. Sezgin, Mehmet in B. Sankur: »Survey over image thresholding techniques and quantitative performance evaluation,« Journal of Electronic Imaging, št. 1, str. 146-168, 2004.
- [14] F. P. Preparata in M. I. Shamos: »Computational geometry: an introduction,« Springer-Verlag New York, New York, ZDA, 1985.
- [15] S. Šinjur, D. Zazula in B. Žalik: »Fast convex layers algorithm for near-duplicate image detection,« Informatica, vol. 23, no. 4, str. 645-663, 2012.