

ViDA: Vizualizácia distribuovaných algoritmov

Michal Anderle^{1*}

Ján Hozza^{1†}

Školiteľ: Jakub Kováč^{2‡}

¹ Katedra informatiky, FMFI UK, Mlynská Dolina 842 48 Bratislava

² Katedra informatiky, FMFI UK, Mlynská Dolina 842 48 Bratislava

Abstrakt: Koník je úžasný

Kľúčové slová: vizualizácia, distribuované algoritmy

1 Úvod

V zimnom semestri sme absolvovali predmet Úvod do distribuovaných algoritmov a veľmi nás zaujala prednášaná téma. Napadlo nás, že táto téma, aj keď je zložitá, by sa dala vysvetliť bežnému človeku, ktorý má o ňu záujem bez toho, aby si musel študovať zložito písané knihy. Tak vznikol náš ročníkový projekt.

Ako cieľ sme si stanovili nielen zvizualizovanie niektorých dobre známych algoritmov, ale taktiež vytvorenie istej interaktivity s používateľom, ktorá by mu pomohla prehĺbiť a lepšie si zapamätať novonadobudnuté znalosti. Myslíme si totiž, že v tejto oblasti existuje mnoho pekných úvah a trikov, ktoré je dobré si osvojiť. A formou interaktívnej vizualizácie, by sme chceli pomôcť vo výučbe a spoznávaní týchto algoritmov, ale takisto umožniť užívateľovi, aby si sám vyskúšal naprogramovať niektoré z algoritmov. Toto by malo preklenúť medzeru medzi porozumením a schopnosťou aplikácie.

Medzi algoritmy, ktoré sme zatiaľ zvizualizovali patrí distribuované prehľadávanie do šírky a taktiež voľba šéfa na úplnom grafe. Zamerali sme sa však na to, aby užívateľ bol schopný sám vytvoriť algoritmus, ktorý mu naša aplikácia zvizualizuje.

Zvyšok článku je organizovaný nasledovne: v sekcii 2 sa zameriame na samotnú implementáciu, ako funguje a prečo sme zvolili zrovna túto možnosť. V sekcii 3 popíšeme konkrétne algoritmy, ktoré vizualizujeme a v sekcii 4 spomenieme naše plány do budúcnosti.

2 Implementácia

Projekt ViDA vznikol ako ročníkový projekt a nevychádzal zo žiadnych predošlých prác alebo projektov. Celý výzor sme navrhli sami po konzultácii s našim vedúcim. Ako to vyzerá, môžete vidieť na prvom obrázku. Naším prvoradým cieľom, bolo zabezpečiť prehľadnú vizualizáciu a tiež jednoduché používanie aplikácie.

2.1 Vizualizácia

Pri distribuovaných algoritmoch sa bavíme o sieti počítačov, ktoré medzi sebou komunikujú – posielajú si medzi sebou správy. Preto bolo prirodzené, aby sme si reprezentovali počítače a ich vzájomné zapojenie do siete ako graf. Každý vrchol je jeden počítač (proces) a správy sa presúvajú medzi počítačmi, len po hranách v danom grafe.

Keďže samotná sieť počítačov má veľký vplyv na vykonávanie daného algoritmu, v prvej fáze, bolo treba umožniť užívateľovi si pohodlným spôsobom vytvárať a editovať graf. Na to je určená hlavná plocha, kde v dobe keď nebeží žiaden algoritmus, môže pomocou myši meniť výzor grafu. Na spodnej lište je niekoľko tlačidiel, ktoré umožňujú užívateľovi prepínať si medzi možnosťami pridávanie, odstraňovanie alebo pohybovanie. S vybraným módom, môže užívateľ vykonávať danú operáciu na grafe. Takisto, existuje pár známych a často používaných typov grafov, ktoré si užívateľ môže zvoliť priamo aj s danou veľkosťou grafu.

Samozrejme najdôležitejšia časť je vizualizácia samotného algoritmu. Bolo dôležité aby to bolo čo najprehľadnejšie a zároveň to dávalo dostatok informácií. Z viacerých možností, ktoré sme skúšali sme nakoniec vybrali bublinkový interface, kde všetky podstatné informácie sú zobrazované vo vyskakovacích bublinkách.

Počas algoritmu jednotlivé procesy často označujú nejakú informáciu, aby dali najavo, čo sa v nich deje. Zo začiatku sme tieto informácie zobrazovali v okienku napravo, aby sme nezahltili priestor grafu zbytočnými textami. Toto všeobecné okienko

*zaba@ksp.sk

†janoh@ksp.sk

‡kuko@ksp.sk

bolo však mimo grafu a bolo skoro nemožné sledovať čo sa deje vnútri grafu (kde ide aká správa) a zároveň si dávať pozor, čo vraví ktorý vrchol. Preto sme sa rozhodli, že informácie vrchola sa budú zobrazovať priamo pri vrchole vnútri grafu, čo sa nakoniec ukázalo, že nie je až tak zavádzajúce, hlavne ak tieto správy časom miznú. Ponechali sme aj panel naboku, ktorý má slúžiť, keď si chce užívateľ pozrieť, čo sa dialo v histórii. A tento panel nezavádza, keďže je decentne skrytý a ukáže sa až na užívateľov príkaz.

Pri distribuovaných algoritmoch nás zaujíma hlavne to, ako sa algoritmus presúva z jedného stavu do druhého na základe doručenej správy. Pod pojmom stav rozumiem naplnenie niektorých dôležitých premenných. Je teda zjavné, že je dôležité, aby užívateľ videl (alebo aspoň tušil), akú hodnotu majú tieto dôležité premenné. Dôležité premenné sú napríklad ID vrchola, či je živý alebo nie, jeho level ... Vypisovať tieto premenné v bublinkách pri vrchole, by však bolo zavádzajúce, keďže je to informácia, ktorá je potrebná stále. Preto sme zvolili taký prístup, aby výzor vrcholu reprezentoval dané premenné.

Keďže ID vrchola je jedna z najdôležitejších informácií, lebo je nezávislá od algoritmu, ktorý je spustený a teda často slúži na prelomenie určitej symetrie, táto hodnota sa vypisuje priamo vnútri vrchola. Ďalšie atribúty vrcholu sú jeho farba, alebo veľkosť. Preto naše vizualizácie často využívajú tieto vlastnosti a intuitívne ich spájajú s nejakou premenou. Napríklad mŕtvy proces zmení svoju farbu na červenú, alebo zväčšujúci sa level zväčšuje veľkosť vrchola.

Samozrejme, občas je dôležité aby sa užívateľ mohol pozrieť na skutočnú hodnotu danej premennej. Preto si vie označiť vrchol, ktorý mu vo vedľajšom okienku ukáže premenné daného procesu.

Podakovanie

Nedakujeme FMFI UK za podporu.