**Uvod>**

V teto praci jsem resil upravu zpusobu agregace dat v grafove databazi pomoci metod zpracovani dat znamych z proudovych systemu za ucelem otestovat, zda upravami dojde k urychleni vykonavani dotazu.

Agregaci zde myslim casti dotazu Group by a Order by.

Samotna uprava mela za cil docilit schopnosti agregovani dat jiz v prubehu vyhledavani vzoru misto aby dochazelo k agregaci klasickym zpusobem az v poslednim kroku vykonavani dotazu po ziskani vsech dat prohledavani.

**Jak jsem toho docilil>**

K realizaci prace jsem implementoval vlastni dotazovaci engine pracujici se statickou grafovou databazi vyuzivajici labeled property grafovy model. Dotazovaci engine pracuje s vybranou podmnozinou jazyka PGQL -> obsahuje zakladni vyrazy zname z SQL… select, match, order by, group by. (mozna tam dat slajd s prikladem?)

Dotazovaci engine jsem implementoval tak aby dokazal provadet agregace klasickym pristupem po dokonceni vyhledavani vzoru i zminenym zpusobem. Pro upraveny i klasicky pristup jsem implementoval jednovlaknova I paralelni reseni.

**Klasicky pristup>**

Dotaz je vykonan skvencnim provadenim hlavnich casti dotazu (tj. Group by, select, match) dle priority. Nejnizsi prioritu ma select a je vykonan jako posledni krok a match ma nejvyssi a je vykonan jako prvni.

Prohledavani grafu probiha pomoci algoritmu DFS, ktery je spusten z kazdeho vrcholu grafu. Vicevlaknove reseni paralelizuje spousteni DFS z nekolika vrcholu soucasne. Muzume se na to divat jako na instance, kde kazda provadi sve lokalni prohledavani. Vysledky prohledavani jsou ukladany do tabulky a s vysledky se pak pracuje pomoci proxy trid.

Order by cast vyuziva knihovni algoritmus merge sort a paralelni verze taktez.

Group by je vykonavan pomoci slovniku/mapy. Pro paralelni pristupy jsem zvolil 3 metody.

1. Dochazi ke globalnimu zpracovani a vlakna sdileji jednu paralelni mapu.
2. Vlakna pracuji vzdy lokalne a nasledne dojde k dvoucestnemu slevani.
3. V tretim pristupu vlakna nejdrive pracuji pouze lokalne a nasledne dojde ke globlanimu slevani misto dvoucestneho.

**Samotna uprava dle zadani>**

Zde budu obecne mluvit rovnou o paralelnich resenich a budu uvazovat, ze jednovlaknove je pouze paralelni reseni s jednim vlaknem.

Instance prohledavani grafu nyni neukladaji vysledky do tabulky, ale preposilaji je k dalsimu zpracovani, tedy tvori z meho pohledu abstaktni proud. Na konci proudu lezi instance trid pro zpracovani group by a order by.

Obecne jsem vymyslel dva hlavni pristupy ktere zde testuji

1. Globalni zpracovani, tj po dokonceni prohledavani je zaroven dokoncena I agregace, sdilena struktura
2. Lokalni zpracovani, po dokonceni prohledavani musi dojit ke slevani, pomalejsi ale mozna efektivnejsi protoze neni treba synchronizace

Order by

Pro zatrizovani u order by jsem vyuzili obecny ab strom a upravenou verzi ktera prvky se stejnymi hodnotami shlukuje do pole misto aby je vlozila do stromu. (zminka, ze ten vysledek se ulozi do tabulky a pak se zatrizuje pouze index z tabulky)

V 2. Lokalni zpracovani dochazi dochazi k kazde vlakno zpracovava jeden data v jednom proudu a nasledne se veci slevaji pomoci paraleniho dvoucestneho mergesortu

V 1. Jsme rozsah hodnot typy podle ktere se tridi rozdelili na mensi intervaly a pro kazdy interval vytvorili prihradku, ktera je pristupna skrze zamek

Group by

Zde jsem upravil jiz zminene pristupy z puvodniho reseni group by. Kazdy proud je zpracovavan lokalne, vysledek po zde jsem zkusil ponech jeste puvodni zneni, ze jsou ukladany do puvodni tabulky a pak se pracuje s proxy tridou, vysledek se kopiruje do tabulkky pouze v pripade, kdy jeste pro nej neexistuje skupina.

V druhem globalnim pristupu jsem implementoval reseni, ktere nepouziva tabulku vysledku ale rovnou pracuje s vysledkem. Proc? Protoze je zde paralelni mapa a bylo by problematicke provadet upravy nad tabulkou protoze jsou zde veci globalni.

**Testovani a vysledky>**

Vsechna navrzena reseni jsme testovali na 3 realnych grafech s umele vygenerovanymi vlastnostmi. Testoval jsem na mem osobnim notebook a vysledky reprezentoval jako median vsech ziskanych hodnot. Nyni bych chtel ukazat nejdulezitejsi vysledky

Graf 1 –

Znacne zrychleni nastalo v pripade parallenich reseni order by pri trideni pomoci hodnot vlastnosti. Na obrazku vidime modre puvodni paralelni merge sort a zbytek jsou navrzena paralaleni reseni. Nejrychlejsi reseni prekonava merge sort az o 38 procent. Q3 jsou hodnoty nahodnych integer a Q4 nahodne retezce

Graf 2 –

K dalsimu zrychleni nastalo pri jednovlaknovych resenich group by. Kdy verze half streamed byla pouze mirne rychlejsi a jinak vynikalo streamed reseni nevyuzivajici tabulky vysledku. Zrechleni zde bylo v rozmezi 13 procent. Napravo vidime, podle ceho byly vysledky seskupovany.