سومین کنفرانس فناوری اطلاعات و دانش
۶ تا ۸ آذرماه ۱۳۸۶
IKT 2007 کنفرانس بین المللی فناوری اطلاعات و دانش
دانشگاه فردوسی مشهد

ترکیب الگوریتم ژنتیک با الگوریتم تپه نوردی در ماشین با ساختار موازی برای بهینه سازی پرس و جوهای بزرگ در گراف پرسش

سعید پارسا، حبیب ایزدخواه و امیر حسین زاده

آدرس محل کار : دانشگاه علم و صنعت ایران، دانشگاه آزاد واحد عجب شیر و دانشگاه آزاد واحد خامنه

E-mail:parsa@iust.ac.ir, habib_eizadkhah@yahoo.com , amirhosseinzade@yahoo.com

چکیده - بهینه‌سازی پرس و جو یک فرآیند گران و پرهزینه می‌باشد و تعداد جایگشت های مختلف برای یک پرس و جو بصورت نمائی با تعداد روابط سهیم در پرس و جو رشد می‌کنند.تکنیکهای فعلی بهینه سازی پرس و جو برای پشتیبانی بعضی از کاربرد های ناشی از پایگاه داده که دارای پیوندهای مختلفی از انواع پرس و جو ها می باشد ، نامناسب هستند از طرفی رمز موفقیت یک مدیر پایگاه داده (*DBMS*) کارا بودن مدل پرس و جوی آن می باشد. دراین مقاله ما پرس و جو هائی را بررسی می کنیم که دارای تعداد روابط زیادی بوده و همچنین گراف آنها دور داشته باشند. که این چنین پرس و جو هائی زمان زیادی را برای پاسخگوئی به پرس وجو نیاز دارند. ما از دو روش برای بهینه سازی این چنین پرس و جوها استفاده می کنیم . ابتدا، بر اساس الگوریتمی گرافها را به چندین درخت تجزیه نموده و سپس درختها را روی کامپیوترها توزیع و با استفاده از ترکیب الگوریتم ژنتیک با الگوریتم تپه نوردی به بهینه سازی آنها پرداختیم و در روش دوم بدون تجزیه گراف به درخت آن را روی کامپیوترها توزیع کرده و به بهینه سازی آن با استفاده از ترکیب الگوریتمهای ذکر شده پرداختیم. پس از ده بار اجرای الگوریتم و از روی نتایج بدست آمده مشخص شد که در روش اول بیشتر از روش دوم جواب بهینه به دست می آید.

کلید واژه- پایگاه داده, بهینه سازی پرس و جو, الگوریتم ژنتیک, گراف پرسش, ماشین موازی

۱- مقدمه

رمز موفقیت یک DBMS به ویژه اگر با مدل رابطه ای کار داشته باشیم ، کارا بودن مدل پرس و جوی بهینه سیستم می باشد . ورودی در این سیستم، یک پرس و جو مانند q به DBMS توسط کاربر می باشد .

فرض کنید S مجموعه همه استراتژی های ممکن برای پاسخگویی به پرس و جوی q باشد هر عضو S مانند s یک هزینه$C(S)$ را دارد (از نظر CPU و I/O) . هدف یک الگوریتم بهینه سازی پیدا کردن یک عضو s0 از S می باشد به طوری که[13] :

$$C(s_0) = \min_{s \in S} C(s)$$

حال بحث را در مورد پایگاه داده های رابطه ای ادامه می دهیم . یک استراتژی برای پاسخگویی به پرس و جوی q ، دنباله ای از عملگر های جبری برای روابط پایگاه داده به کار می برد تا در نهایت جواب برای q را بدست آورد . هزینه یک استراتژی مجموع هزینه های پردازش هر عملگر می باشد . در میان این عملگر ها مشکل ترین آنها برای پردازش و بهینه سازی ، عملگر پیوند (که با علامت $\bowtie$ یا JOIN نشان داده می شود) می باشد . این عملگر دو رابطه را به عنوان ورودی دریافت می کند سپس با ترکیب تاپل ها (tuple)و با توجه به نوع پیوند ، یک رابطه جدید به عنوان خروجی تولید می کند. پیوند ، یک عملگر جابجا پذیر و شرکت پذیر می باشد . لذا تعداد استراتژی ها برای پاسخگویی به یک پرس و جو به صورت نمایی با تعداد پیوند ها رشد می کند[15][12][4][3][2] . به عنوان مثال اگر چه دو عبارت

(R1 JOIN R2) JOIN R3=R1 JOIN (R2 JOIN R3)معادلند و خروجی یکسانی دارند ولی هزینه زمانی آنها می تواند یکسان نباشد، زیرا کاردینالتی رابطه میانی که باید ذخیره