ارائه یک روش جدید بهینه سازی پرس و جو در پایگاه داده مشبک

پیمان عاربی[۱]، سراج الدین کاتبی[۲]، امیر مسعود بیدگلی[۳]

۱-گروه مهندسی کامپیوتر دانشگاه آزاد اسلامی واحد علوم و تحقیقات اهواز،peymanarebi@gmail.com
۲- عضو هیئت علمی دانشگاه آزاد اسلامی واحد بوشهر،Katebi@shirazu.ac.ir
۳- عضو هیئت علمی دانشگاه آزاد اسلامی واحد تهران شمال،Drbidgoli@gmail.com

چکیده:

پایگاه داده مشبک به منظور ایجاد ساختاری جهت ذخیره سازی داده ها در عرض محیطی مشبک به صورت توزیع شده و ناهمگن سعی در ذخیره سازی و پراکندگی داده در محدوده جغرافیای وسیعی دارد. به دلیل وسعت و حجم بالای داده پردازش تراکنش در چنین محیطی بسیار پیچیده و زمانبر می باشد. بدیهی است در صورت استفاده از پرس و جوها بدون بهینه سازی، کارایی تراکنشها را در این پایگاه داده به شدت پایین می آورد، در صورتیکه با استفاده از الگوریتمهای بهینه سازی مناسب می توان تا حد زیادی کارایی را افزایش داد. الگوریتمهای زیادی برای بهینه سازی پرس و جو ها پیشنهاد شده است اما به دلیل متفاوت بودن محیطهای مشبک نیاز به الگوریتمهای بهینه سازی متفاوتی می باشد. در این مقاله الگوریتمی ارائه شده است که منطبق بر ساختار محاسبات مشبک بوده و در سیستمهای پایگاه داده مشبک با حجم داده بالا به خوبی عمل می کند.

کلمات واژه: محیطهای مشبک، پایگاه داده مشبک، الگوریتمهای بهینه سازی، توزیع پذیری، گلوبوس، بهینه سازی پرس و جو،زمان پاسخ، خودمختاری، ناهمگن

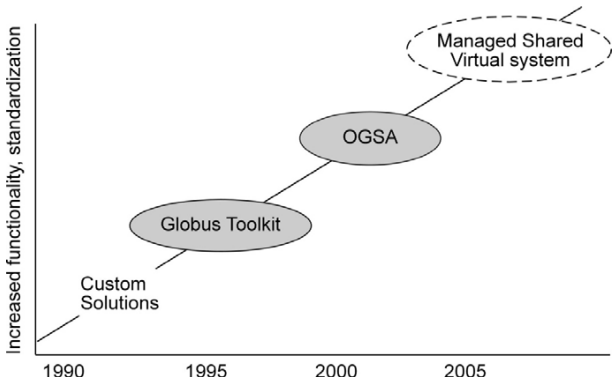
۱- مقدمه

با افزایش حجم داده ها در پایگاه داده مربوط به سازمانها به خصوص سازمانهایی که دارای چند بخش و پراکنده از لحاظ جغرافیایی هستند، مشکلات فراوانی را جهت ذخیره ، بازیابی و انتقال داده به وجود آورده است. به طور مثال در یک پایگاه داده با حجم ۱۰ ترابایت در صورتیکه ساده ترین پردازش را بخواهیم انجام دهیم، در این صورت این پردازش در حدود ۱۲۰ شبانه روز طول خواهد کشید. از طرفی تجمع داده ای مربوط به پایگاه داده موجود در چند مکان مختلف جغرافیایی از لحاظ امنیت و سرعت اجرای تراکنشها پر خطر می باشد و عموما دارای ضعفهای فراوانی است؛ که این مشکلات با استفاده از پایگاه داده مشبک قابل حل و از لحاظ عملکرد موثرتر می باشد. از طرف دیگر با گسترش شبکه اینترنت در بین سازمانها و موسسات و منطبق بودن معماری مشبک بر این شبکه، ناگزیر

به توسعه پایگاه داده ای در محیطهای مشبک هستیم.از طرف دیگر انجام تراکنشها در محیطهای پایگاه داده به طور مستقیم به سرعت اجرای پرس و جوها وابسته است. از آنجا که در محیطهای مشبک حجم داده و فواصل ارسال داده زیاد می باشد سرعت اجرای پرس و جوها عموما پایین می باشد. راه حل نخست ارتقاء سخت افزارهاست که این امر هزینه بر و دارای محدودیتهایی است، اما راه حل دیگر بهینه سازی پرس و جوها با استفاده از الگوریتمهای مناسب می باشد.

۲- محیطهای مشبک[1]

معماری مشبک در ابتدا یکی از روشهای اتصال شبکه ارتباطی در محاسبات موازی بود اما به تدریج تبدیل به یک معماری مستقل و پرکاربرد شد. این ساختار به دلیل ایجاد معماری میان افزار (معروفترین آنها گلوبوس[2]) جهت ارتباط بین منابع، دارای قابلیتهای زیادی در ارتباط داده ای می باشد.در یک محیط مشبک برنامه های کاربردی دسترسی، مدیریت و توزیع داده را در مقیاس بزرگ در عرض چند سازمان ایجاد کرده و با تقسیم بندی و توزیع داده در مکانهای مختلف جغرافیایی با منابع ناهمگن[3] امکان ذخیره و بازیابی و اجرای تراکنشها را با حجم بسیار بالا و تقریبا نامحدود فراهم می آورد. ویژگی اصلی این محیطها، رشد گسترده داده، توزیع داده و نامتقارن بودن منابع داده ای است. بنابراین بر اساس این معماری می توان پایگاه داده مشبک را با همان ویژگیهای محیطی ایجاد کرد. تکنولوژی سیستم مشبک در طول سالها تحقیق و توسعه هم در قسمت‌های تحقیقاتی و علمی، و هم در زمینه‌های صنعتی پدیدار شده است و اکنون نیز ادامه می‌یابد. همانطور که شکل ۱ نشان می‌دهد، چهار فاز را برای تکامل محاسبات مشبک در نظر گرفت.



شکل ۱: سیر تکامل تکنولوژی Grid

[1] Grid Environments
[2] Globus
[3] Heterogeneous

۳- پایگاه داده مشبک[4]:

تکنولوژی مشبک در دهه های اخیر مورد توجه بسیاری از محققان قرار گرفته است اما اخیرا مدیریت داده در محیطهای مشبک نظیر مشبک داده ای[5] و پایگاه داده مشبک نیز توجه محققان را به خود جلب کرده است. این مورد به دلیل وجود معماری میان افزار[6] مانند گلوبوس و لجیون و غیره می باشد. شکل ۲ معماری کلی یک پایگاه داده در یک محیط مشبک را نشان می دهد

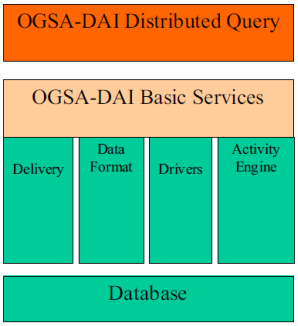
شکل ۲ : زیر معماری کلی یک پایگاه داده در یک محیط مشبک

یک سیستم پایگاه داده در محیط مشبک دارای سه بعد می باشد:

خودمختاری ۲- توزیع ۳- ناهمگن

معماری پایگاه داده مشبک به دو شکل پایگاه داده موازی و پایگاه داده توزیع شده قابل پیاده سازی است. در پایگاه داده موازی از سخت افزارهایی جهت ایجاد محیطی موازی با به هم مرتبط کردن پردازنده ها به شکل مشبک استفاده می شود. این ساختار عموما ساختاری پرهزینه و پیچیده می باشد. اما ساختار دیگری که امروزه بیشتر مورد استفاده قرار می گیرد، معماری پایگاه داده توزیع شده می باشد. این ساختار مجموعه ای از پایگاه داده ای است که به طور منطقی با هم به اشتراک گذاشته شده اند و به طور فیزیکی در عرض چند گره جداگانه در یک شبکه توسط یک سیستم مدیریت پایگاه داده مدیریت می شود به طوریکه از دید کاربران، به عنوان یک سیستم پایگاه داده در نظر گرفته می شود. در این سیستم اشتراک گذاری داده به طور فیزیکی نیست و هر یک از سیستمها درون یک سایت مستقل ایجاد می شوند. از آنجاییکه این معماری منطبق بر معماری اینترنت می باشد، در حال حاضر پرکاربردتر از معماری موازی می باشد. در این مقاله از معماری توزیع شده پایگاه داده مشبک استفاده شده است.

یکی از معروفترین و کاملترین معماریهای پایگاه داده مشبک ارائه شده، معماری OGSA-DAI می باشد. هدف اصلی این معماری ایجاد یک سرویس یکپارچه[7] واسط برای دسترسی داده و مجتمع سازی پایگاه داده تشکیل دهنده مشبک و پنهان سازی نوع پایگاه داده و فرمت داده ای و همچنین مکانیزم پاسخ دهی از دید کاربران می باشد. OGSA-DAI دارای منابع داده ای نامتجانس و ناهمگن می باشد که می توانند سرویسهای OGSA را یکپارچه کنند. در شکل ۳ اجزاء کلی این معماری را مشاهده می کنید.



شکل ۳ :اجزاء کلی معماری OGSA-DAI

[4] Grid Database

[5] data Grid
[6] middleware
[7] uniform service

تراکنشها و اجرای پرس و جو[8]

پرس و جو ها در یک سیستم پایگاه داده مهمترین ابزار در ارتباط داده ای بین برنامه های کاربردی و همچنین تراکنشهای درون پایگاه داده می باشند. در ایجاد و اجرای پرس و جو مسیرهای مختلفی جهت دسترسی به منابع داده ای وجود دارد. اجرای پرس و جو در یک محیط مشبک در مقایسه با محیطهای سنتی متفاوت می باشد. در اینگونه محیطها برنامه های کاربردی در ارتباط با پایگاه داده مشبک به منظور اجرای پرس و جو از ساختار زیر استفاده می کنند.

Application(Work(),Data(),Time(),Geography(),Query())

که در این ساختار:

Work: مشخص کننده اجرای دسته ای[9] و تعیین همزمانی[10] در شروع کارهاست.

Data: مشخص کننده نوع و حجم داده ارسال شده

Time: مشخص کننده دوره زمانی ارسال داده

Geographic: مشخص کننده مکان ارسال داده و یا درخواست کننده داده

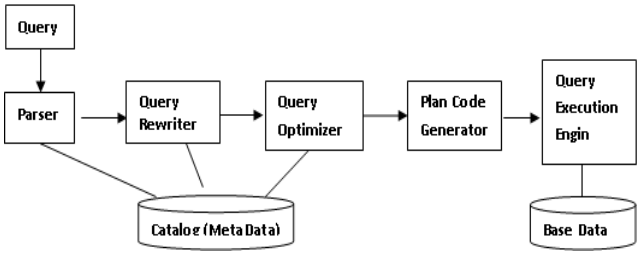
Query: پرس و جو مورد نیاز جهت عملیات پایگاه داده که به دو صورت ساده و پیچیده می باشد.

پرس و جو ها مهمترین تاثیر را در سرعت دسترسی به داده ها در تراکنشهای پایگاه داده ای دارند. در صورتیکه پرس و جو ها با سرعت بالایی اجرا شوند از یک طرف حجم بالاتری از داده ها را پردازش می کنند و از طرف دیگر زمان اجرای یک تراکنش را تسریع می بخشند. افزایش سرعت یک پرس و جو معلول عوامل زیادی است. برخی از این عوامل بستگی به سخت افزار و معماری محیطی می باشد و برخی دیگر مربوط به ساختار آن پرس و جو می باشد.

بهینه سازی پرس و جو[11]:

در یک سیستم پایگاه داده ای دو نوع پرس و جو وجود دارد. پرس و جوی ساده و پیچیده. در پرس و جوهای ساده هدف، پردازش یک رابطه است در صورتیکه در پرس و جوهای پیچیده چندین رابطه ترکیبی پردازش می شوند. در بهینه سازی پرس و جو ها نوع دوم بیشتر مورد توجه قرار می گیرد.

یک پرس و جو قبل از اجرا پویش[12] شده و مورد تجزیه و تحلیل قرار می گیرد. پس از مرحله تجزیه و تحلیل، پرس و جو جهت اجرا ارسال می شود. فرآیند بهینه سازی باید در بین این دو مرحله یعنی پس از تجزیه و تحلیل و قبل از اجرا انجام شود. در شکل ٤ این مراحل را مشاهده می کنید.



شکل ٤: روال بهینه سازی در اجرای پرس و جو

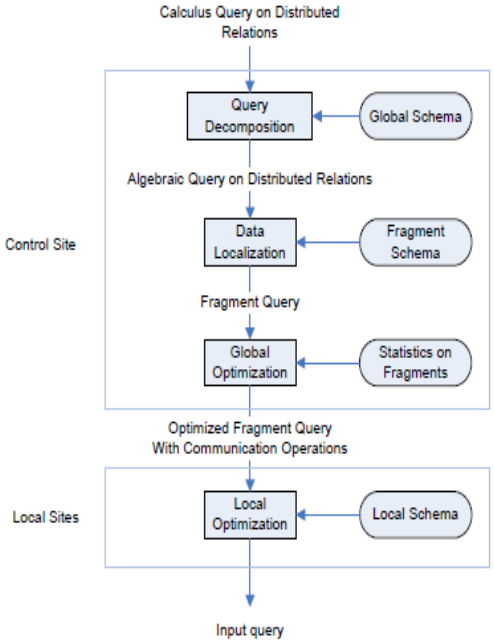
[8] Query
[9] batch
[10] synchronous
[11] Query Optimization
[12] Scanning

۵-۱ معماری بهینه سازی پرس و جو:

همانطور که در بحث معماری پایگاه داده مشبک گفته شد، پردازش پرس و جو در دو مرحله انجام می شود. معماری بهینه سازی پرس و جو نیز منطبق بر این دو مرحله می باشد.

ساختار این معماری را در شکل ۵ مشاهده می کنید.

این معماری از دو فاز تشکیل شده است:



شکل۵: معماری بهینه سازی پرس و جو در پایگاه داده مشبک

۵-۱-۱بهینه سازی سراسری۱۳:

در این مرحله بهینه سازی در سطح شبکه سراسری مشبک صورت می گیرد، بدین صورت که پارامترهای مربوط به ارتباط درون شبکه ای و ساختارهای ارتباطی محیط مشبک، بهینه سازی می شود.

۵-۱-۲ بهینه سازی محلی۱٤:

در مرحله بعد پس از آنکه پرس و جو به گره مورد نظر در محیط مشبک ارسال شد، بار دیگر پرس و جو به صورت محلی مورد بازبینی و بازنویسی قرار گرفته و بر اساس ساختارها و پارامترهای محلی بهینه سازی می شود.

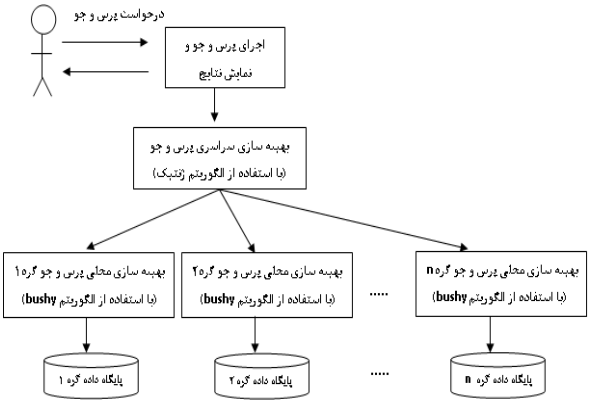
۵-۲ پارامترهای مورد استفاده:

درمحیطهای مشبک جهت برآورد سرعت اجرای پرس و جو از دو پارامتر بر اساس تعداد گره و همچنین حجم داده استفاده می شود.

[13] Global optimization
[14] Local Optimization

زمان پاسخ[15]:
مدت زمان ارسال پرس و جو برای اجرا تا زمان دریافت پاسخ
نرخ تسریع[16]:
نسبت سرعت اجرای پرس و جو پس از بهینه سازی به قبل از بهینه سازی

۵-۳ معماری پیشنهادی بهینه سازی پرس و جو:
معماری پیشنهادی بر اساس معماری بهینه سازی پرس و جو ایجاد و بر اساس ویژگیهای مراحل ذکر شده از الگوریتمهای مناسب با کارایی بالا استفاده شده است.
ساختار معماری پیشنهادی را در شکل ٦ مشاهده می کنید



شکل٦ : معماری پیشنهادی بهینه سازی پرس و جو

در این معماری در مرحله اول از یک الگوریتم ژنتیک جهت بهینه سازی سراسری استفاده شده و در مرحله دوم پرس و جو به چند زیر پرسو جو تقسیم شده و هرکدام جهت اجرا به گره مربوطه ارسال می شود. در گره های مربوطه نیز از الگوریتم Bushy متوازن جهت بهینه سازی محلی استفاده می شود.

٦- الگوریتم پیشنهادی بهینه سازی پرس و جو:
الگوریتم ژنتیک جهت بهینه سازی سراسری:
در بهینه سازی سراسری سعی بر آن است که بر اسا س ساختار شبکه گریدی و هزینه ارتباط بین گره ها و نحوه اتصال آنها به هم جداول مورد استفاده در پرس و جو را به گونه ای انتخاب کنیم که کمترین هزینه ارتباط را در شبکه مشبک داشته باشد. جهت تعیین ترتیب بازیابی و جستجوی جداول در محیط مشبک از یک الگوریتم ژنتیک استفاده شده است. بنابراین می بایست در ابتدا بخشهای مختلف الگوریتم ژنتیک که عبارت است از کروموزومها ، عملگرها و تابع شایستگی الگوریتم مشخص شود.
کروموزومها:
در این الگوریتم برای تعیین کروموزومها از ترتیب جداول مربوط به گره ها استفاده شده است. مثلا اگر پرس و جویی به شکل زیر داشته باشیم

[15] Response Time
[16] SpeedUpRate

R1∞R2∞R2∞R4∞R5

به طوریکه شماره روابط، نشان دهنده شماره گره مربوطه باشد، کرموزومها به شکل ترتیبی از جایگشتهای شماره گره ها در نظر گرفته شده است. مثلا 13254 و یا 35124 و ....

عملگرها:

در الگوریتم ژنتیک دو عملگر اصلی داریم، عملگر کراس آور که وظیفه انتخاب کروموزومها و عملگر میوتیشن که وظیفه ترکیب کرووزومها به منظور جهش کرموزومی برای رسیدن به جواب مناسبتر را بعهده دارد. در این الگوریتم عملگر کراس آور به صورت نقاط رندوم تعریف شده است و ترتیب شماره گره ها حول این نقاط چرخش می کنند. در این الگوریتم از الگوریتم CHUNK استفاده شده است. این عملگر به این صورت عمل می کند که ابتدادو کرموزوم که بالاترین فیتنس را دارند انتخاب شده، مثلا ترتیب زیر:

X=1 5 2 3 4

Y=3 5 1 4 2

با فرض اینکه طول کرموزوم L باشد الگوریتم CHUNK در ابتدا عددی رندوم در بازه [0 , L/2] انتخاب را انتخاب کرده سپس عدد رندوم بعدی در بازه [L/2,L/4] انتخاب شده و تا زمانیکه طول بازه کمتر از صفر نشده ادامه پیدا می کند. اعداد رندوم ایجاد شده که تعداد آنها نیز L می باشد جایگاه ژنها را در ترتیب کرموزومی مشخص می کنند. جهت جلوگیری از ایجاد ژن تکراری از یک تابع برای کنترل این مورد استفاده می شود. مثلا یک کرموزوم جدید حاصل از این عملگر ممکن است به شکل زیر باشد

Z=5 1 2 3 4

تابع شایستگی[17]:

تابع شایستگی در الگوریتم مشخص کننده درصد پذیرش پاسخ مناسب در بین کرموزومهاست. این تابع امکان تشخیص جواب مناسب و یا روند رسیدن پاسخ بهینه را نشان می دهد. چون این الگوریتم در سطح سراسری شبکه مشبک جهت کاهش زمان پاسخ از طریق کاهش هزینه ارتباط صورت می گیرد، بنابراین در تابع شایستگی مینیمم هزینه ارتباط در اجرای کرموزومهای حاصل از یک پرس و جو مد نظر می باشد.

Fitness=Minimum{CommCost(chromosome(i)),

}CommCost(chromosome(i-1))

شبه کد اجرایی این الگوریتم را در شکل ۷ زیر مشاهده می کنید.

الگوریتم Bushy متوازن جهت بهینه سازی محلی:

در مرحله دوم پس از آنکه با استفاده از الگوریتم مرحله قبل گره مورد نظر جهت اجرای پرس و جو انتخاب شد، زیر پرس و جوی ارسال شده به گره با استفاده از الگوریتم Bushy متوازن بهینه سازی و اجرا می شود. این الگوریتم بر اساس توازن درخت پرس و جوی ایجاد شده پس از مرحله پارس کردن پرس و جو عمل می کند. در این الگوریتم ابتدا درخت پرس و جو با استفاده از الگوریتم توازن در صورت عدم توازن، متوازن شده و سپس با استفاده از الگوریتم Bushy فضای داده ای مورد جستجوی محدود تر می شود. به طور مثال در عمل پیوند که یکی از پرهزینه ترین عملگرهای پرس و جو می باشد این الگوریتم بدین شکل عمل می کند.

[17] Fitness Function

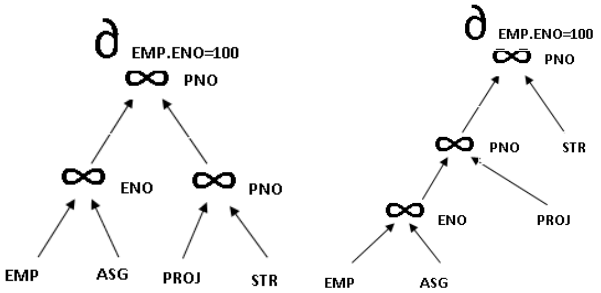
```
Input:List of Relation and Node Number in Query
Output: Optimization Order of Relation in Query
Repeat
        For each choromosome i do
                (1).Evaluate f(i)
                (2).Broadcast f(i) in other order
                (3).Receive f(j)  for all chromosome in another order
                (4).Select chromosome k to Crossover of i base fitness
                (5).Reproduce using chromosome i  and k
                (6).replace chromosome i with one of the offspring
until  population variance is small
```

شکل۷ : شبه کد الگوریتم ژنتیک بهینه سازی سراسری

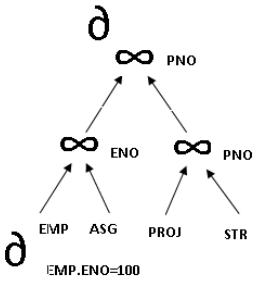
پرس و جوی زیر را در نظر بگیرید

```
SELECT  ENAME,RESP
FROM    EMP,ASG,PROJ,SRT
WHERE   EMP.ENO=ASG.ENO  AND
ASG.PNO=PROJ.PNO
AND   PROJ.PNO= SRT.PNO AND  EMP.ENO=100
```

در شکل ۸ درخت پرس و جوی مربوط به این پرس و جو را مشاهده می کنید. در شکل پرس و جوی فوق با استفاده از الگوریتم توازن مورد بازنویسی قرار گرفته و در شکل با استفاده از الگوریتم Bushy این پرس و جو بهینه سازی شده است.



شکل۸ الف: درخت متوازن شده پرس و جو شکل۸ ب :درخت پرس و جو

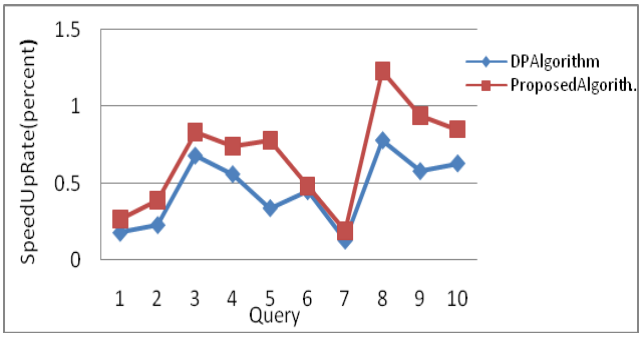


شکل۸ ج: درخت بهینه سازی شده با الگوریتم Bushy

متوازن با ترکیب این دو الگوریتم، یک الگوریتم بهینه سازی با کارایی بالا جهت بهینه سازی پرس و جو در پایگاه داده مشبک ایجاد خواهد شد.

۷- نتایج شبیه سازی

ما در شبیه سازی از یک فضای داده ای با حجم بالا(در حدود ۱ ترابایت) که در یک محیط مشبک با پایگاههای ناهمگن گسترده شده است، استفاده کرده و دو پارامتر بحث شده در بخشهای قبلی را مورد بررسی قرار داده ایم. به منظور مقایسه الگوریتم پیشنهادی با دیگر الگوریتمها و بررسی آن، نتایج حاصل از اجرای این الگوریتم را با الگوریتم برنامه نویسی پویا[18] برای بهینه سازی پرس و جو که یکی از الگوریتمهای مهم در بهینه سازی می باشد و همچنین اجرای پرس و جوهای بدون بهینه سازی را با هم مقایسه کرده ایم. در شکل ۹ نتایج مربوط به نرخ تسریع برای ۱۰ پرس و جو که در آنها از عملگرهای مختلف جبر رابطه ای نظیر پیوند، نیم پیوند، جستجو و حاصلضرب دکارتی استفاده شده است، را برای دو الگوریتم برنامه نویسی پویا و الگوریتم پیشنهادی مشاهده می کنید.



شکل ۹ : مقایسه نرخ تسریع

در جدول۱ پارامترهای مورد استفاده در الگوریتم ژنتیک را مشاهده می کنید.

[18] Dynamic Programming

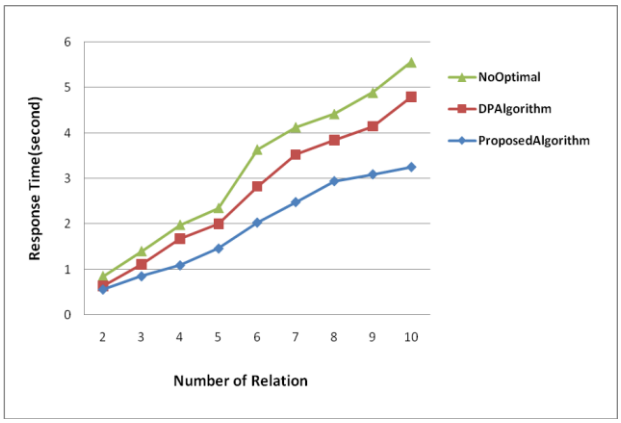
جدول۱- پارامترهای الگوریتم ژنتیک مورد استفاده

پارامترها	مقدار
اندازه جمعیت	۲۰۰
احتمال میوتیشن	۰.۰۱
احتمال کراس آور	۰.٦
احتمال در صد شایستگی	۰.۵
تعداد مراحل تولید	۲۰۰

پارامتر دیگر مورد بررسی زمان پاسخ پرس و جوهاست.در جدول۲ زمان پاسخ مربوط به پرس و جوهایی با تعداد روابط متفاوت برای سه حالت نشان داده شده است. در شکل ۱۰ نیز نتایج حاصل از این مقایسه را نشان می دهد.(k نشان دهنده تعداد روابط در پرس و و جو است)

جدول۲- زمان پاسخ مربوط به پرس و جوهای در سه حالت

K	No optimal	DP	algorithm Proposed
2	۰.۸۵	۰.٦٤	۰.۵٦
3	۱.٤۰	۱.۱۲	۰.۸۵
4	۱.۹۸	۱.٦۸	۱.۰۹
5	۲.۳۵	۲.۰۱	۱.٤٦
6	۳.٦٤	۲.۸۳	۲.۰۳
7	٤.۱۳	۳.۵۳	۲.٤۸
8	٤.٤۲	۳.۸۵	۲.۹٤
9	٤.۸۹	٤.۱۵	۳.۰۹
10	۵.۵٦	٤.۸۰	۳.۲۵



شکل۱۰ : مقایسه زمان پاسخ مربوط به سه حالت (بدون بهینه سازی، بهینه سازی با الگوریتم پویا، بهینه سازی با الگوریتم پیشنهادی)

۸- بررسی نتایج

همانطور که نمودارهای نرخ تسریع و زمان پاسخ در نتایج حاصل از شبیه سازی نشان می دهد، در پرس و جوهایی که تعداد روابط کم می باشد الگوریتمهای بهینه سازی نسبت به حالت بدون بهینه تفاوت چندانی در سرعت اجرا ندارند اما با رشد تعداد روابط تفاوت زیادی را در جهت کاهش زمان اجرا و افزایش کارایی در الگوریتمهای بهینه سازی مشاهده می شود. همچنین الگوریتم پیشنهادی که یک الگوریتم هیوریستیک می باشد به مراتب بهتر از الگوریتم برنامه نویسی پویا که مرتبه ای نمایی دارد، عمل می کند .

۹- کارهای آتی:

پایگاه داده مشبک به دلیل کارایی و کاربرد زیاد در سیستمهایی که حجم داده بالایی دارند همواره مورد توجه محققان بوده است. یکی از این زمینه ها داده کاوی در پایگاه داده مشبک می باشد که به دلیل اهمیت این موضوع و همچنین استفاده از الگوریتمهای مناسب در داده کاوی، الگوریتمهای بهینه سازی پرس و جو در پایگاه داده مشبک می تواند نقش مهمی را در اجرای الگوریتمهای داده کاوی مشبک ایفا کند.

منابع:

[1] David Taniar, Clement H.C. Leung, Wenny Rahayu, Sushant Goel. High-Performance Parallel Database Processingand Grid Databases(2008), A John Wiley & Sons, Inc., Publication
[2] April J. Wells, Grid Database Design(2005), Auerbach Publications
[3] Mario Antonioletti, Malcolm Atkinson, Rob Baxter, The design and implementation of Grid database services in OGSA-DAI(2005)
[4] www.dbgrid.com
[5] Yannis E. Ioannidis, Query Optimization, University of Wisconsin,
[6] MICHAEL DI STEFANO, DISTRIBUTED DATA MANAGEMENT FOR GRID COMPUTER
[7] Yin Chen1, Dave Berry1, Patrick Dantressangle2, Transaction-Based Grid Database Replication
[8] D. E. Goldberg, "The genetic algorithms in search, optimization, and machine learning", New Y7ork: Addison-Wesley, 1989.