تخصیص به نزدیکترین همسایه با حد آستانه نسبی (RTNNA)

کاوه پاشائی ، رضا باصدا ، دکتر مسعود رهگذر
دانشگاه تهران- دانشکده فنی - گروه مهندسی برق و کامپیوتر
{K.Pashaei , R.Basseda }@ece.ut.ac.ir , Rahgozar@ut.ac.ir

چکیده : *در این مقاله استراتژی تخصیص داده پویای جدید برای سیستم های پایگاه داده توزیع شده غیرتکراری به نام الگوریتم RTNNA[1] مطرح گردیده است. این الگوریتم با توجه به تغییر الگوی دسترسی به قطعه‌های داده عمل تخصیص مجدد قطعه‌های داده را انجام می‌دهد. در این الگوریتم قطعه‌های داده به نودی منتقل می شود که در نزدیکی نودهایی قرار دارد که بیشترین دسترسی را به این قطعه داده دارند. این الگوریتم با بوجود آوردن خوشه‌های داده برای سیستمهای پایگاه داده توزیع شده که با بار زیاد و درخواستهای متعدد از سایتهای مختلف در یک شبکه مواجه می‌باشند مناسب می‌باشد. نتایج شبیه‌سازی نشان می‌دهد که الگوریتم RTNNA برای شبکه‌هایی که در آنها قطعه‌های داده به طور مکرر از سایتهای مختلف درخواست می‌شود زمان پاسخ بهتری دارد و برای انتقال قطعه‌های داده در شبکه نیاز به زمان کمتری دارد.*

۱.مقدمه : پیشرفت در تکنولوژیهای شبکه و پایگاه داده در دهه های اخیر منجر به ایجاد سیستم های پایگاه داده توزیع شده گشته است .یک سیستم پایگاه داده توزیع شده مجموعه‌ای از سایتها می باشد که از طریق شبکه به هم متصل شده‌اند که هر کدام از سایت ها پایگاه داده مخصوص به خود دارد اما می توانند با یکدیگر کار کنند بنابراین هر کاربری در هر سایتی می تواند به همه داده های موجود در شبکه دسترسی داشته باشد درست مانند اینکه همه داده ها در سایت کاربر ذخیره شده است.[11]

دغدغه اصلی سیستم های پایگاه داده توزیع شده قطعه‌قطعه کردن[2] و تخصیص[3] پایگاه داده اصلی می باشد واحد قطعه داده می تواند یک فایل باشد که در این حالت موضوع تخصیص همان تخصیص فایل خواهد بود مشکل تخصیص داده یک مسئله از درجه NP می باشد بنابراین نیاز به هیوریستیکهای سریع برای تولید راه حل های موثر می باشد علاوه بر اینها تخصیص بهینه اشیا پایگاه داده به طور شدید بستگی به استراتژی اجرای پرس وجو [4] که به وسیله پایگاه داده توزیع شده پیاده سازی شده دارد. مسئله تخصیص فایل به طور کامل در ادبیات مقالات بررسی شده است که ابتدا توسط Chu آغاز شده است[2] و سپس مدلهای تکراری و غیر تکراری در [3][4], بررسی شده و در [5],[6] برخی مطالعات در زمینه تخصیص فایل پویا انجام شده است.

راه‌حل‌های گوناگونی برای تخصیص داده در سیستمهای توزیعی وجود دارد [1], [4], [5], [6]. در این مقالات قبل از طراحی پایگاه داده تخصیص داده براساس الگوهای دسترسی داده استاتیک یا الگوهای پرس‌وجوی استاتیک انجام می‌گیرد. در محیط استاتیک احتمال دسترسی به قطعه‌های داده هرگز عوض نمی‌شود بنابراین در این محیطها از راه‌حلهای استاتیک استفاده می‌شود در حالیکه در محیط پویا این احتمالات دائماً عوض می‌شود و استفاده از روشهای استاتیک کارایی پایگاه داده را پایین می‌آورد. در [3] یک الگوریتم تخصیص داده پویا برای سیستمهای پایگاه داده غیر تکراری به نام optimal ارائه شده ولی هیچ مدلی برای تحلیل الگوریتم بیان نشده است. در

[1] Relatively Threshold Near Neighborhood Algorithm

[2] Fragmentation
[3] Allocation
[4] Query

[5] الگوریتم Threshold مطرح گردیده که یک الگوریتم تخصیص داده پویا می‌باشد که قطعه‌های داده را با توجه به تغییر الگوهای دسترسی داده بین سایتها منتقل می‌کند و روی توازن بار تمرکز دارد.

هزینه اصلی در اجرای پرس و جو در سیستمهای پایگاه داده توزیع شده هزینه انتقال داده هنگام انتقال یک رابطه در موقع درخواست پرس و جو از یک سایت و انتقال آن از یک سایت متفاوت می باشد . هدف اصلی الگوریتم های تخصیص داده تعیین نسبت دادن قطعه‌های داده به سایتهای مختلف برای کمینه کردن هزینه انتقال داده در اجرای یک مجموعه از پرس‌وجوها می باشد که معادل کمینه کردن زمان متوسط اجرای پرس‌وجو می باشد که اهمیت اصلی در محیط های توزیع شده و پایگاه داده چند رسانه ای دارد. الگوریتمهای توزیع پویای داده ، از آنجائیکه معمولا با توجه به نوع بکارگیری ، دو فاکتور متفاوت از شبکه را برای توزیع داده در پایگاه‌داده‌های توزیعی استفاده میکنند ، معمولا از پیچیدگی بالای برخوردار بوده و مساله از درجه NP محسوب میشوند. در این میان ، تکنیکهای مختلفی جهت کاهش درجه مساله بکار گرفته میشود. از جمله این تکنیکها میتوان به استفاده از Heuristic های مختلف اشاره نمود. رایج ترین نوع Heuristic های بکار رفته ، میتوان به تغییر نوع متریک در الگوریتمهای توزیع داده در پایگاه‌داده‌های توزیعی عادی اشاره نمود.

در این مقاله ما دو روش تخصیص قطعه داده پویا به نامهای تخصیص به نزدیکترین همسایه با حد آستانه نسبی و نسخه اصلاح شده آن را بررسی می‌کنیم این الگوریتم‌ها بر پایه الگوریتم optimal می‌باشند[3] ولی از استراتژی مختلفی برای انتقال داده بین سایتها استفاده می‌کنند.

بقیه مقاله به صورت زیر می باشد : در بخش ۲ ما متد خود را توضیح می‌دهیم، بخش ۳ محیط پیاده‌سازی و نتایج را نشان می‌دهد و در نهایت بخش ۴ نتیجه‌گیری می‌باشد.

۲. روش‌شناسی[5] : الگوریتمNNA حالت خاصی از الگوریتم Optimal می باشد در الگوریتم optimal تمام قطعه های داده با استفاده از یک متد استاتیک بین تمام سایتها توزیع می شود سپس هر نود j برای قطعه داده i الگوریتم optimal را مطابق شکل زیر اجرا می کند.

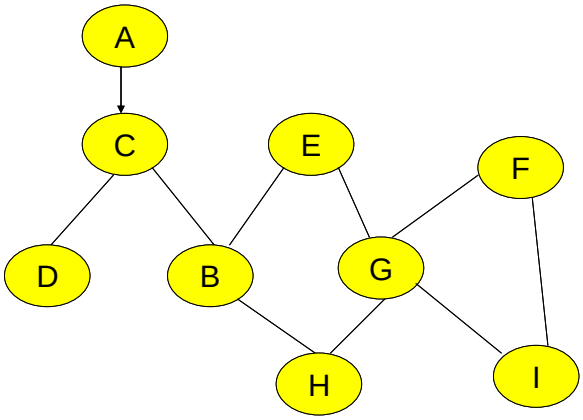
۱. برای هر قطعه داده که به صورت محلی[6] ذخیره شده سطر شمارنده دسترسی را برابر ۰ قرار بده (Sik=0 که k=1,2,…,n)
۲. درخواست دسترسی به قطعه داده ذخیره شده را پراسس کن
۳. شمارنده دسترسی نودی که به این قطعه داده دسترسی پیدا کرده را یکی افزایش بده (اگر نود x به قطعه داده i دسترسی پیدا کند قرار بده six=six+1)
۴. اگر نودی که به آن دسترسی شده همان نود جاری باشد که قطعه داده در آن قرار دارد برو به مرحله ۲ (دسترسی محلی)
۵. اگر شمارنده نود دور[7] بیشتر از نودی باشد که قطعه داده در آن قرار دارد مالکیت این قطعه داده همراه با آرایه مربوط به آن را به نود دور منتقل کن (اگر نود x به قطعه داده i دسترسی پیدا کند و six>sij باشد قطعه داده i را به نود x بفرست)
۶. برو به مرحله ۲

مشکل این الگوریتم (optimal) این است که اگر الگوهای تکرار دسترسی به قطعه های داده زیاد باشد زمان زیادی برای انتقال قطعه های داده به نودهای مختلف صرف می شود بنابراین زمان پاسخ و تاخیر افزایش پیدا می کند در الگوریتم NNA این مشکل حل می شود در الگوریتم NNA شرایط لازم برای اینکه قطعه داده منتقل شود درست مانند الگوریتم optimal می باشد اما مقصد یعنی محلی که قرار است داده به آن منتقل شود فرق می کند در این روش توپولوژی

[5] Methodology
[6] locally
[7] Remote Node

شبکه و مسیریابی[8] برای مشخص کردن مقصد در نظر گرفته می شود به عبارت دیگر مقصد قطعه داده ای که می خواهد منتقل شود نودی می باشد که همسایه نود مبدا است و این نود همسایه یعنی نودی که قرار است قطعه داده به آن منتقل شود در مسیری قرار دارد که نودهای موجود در این مسیر بیشترین دسترسی به این قطعه داده را دارند در این الگوریتم (NNA) از الگوریتم link state برای مسیربابی استفاده شده است با استفاده از این روش انتقال مکرر قطعه های داده به دلیل اینکه این قطعه های داده در نودی قرار می گیرد که برای همه نودهایی که به این قطعه داده دسترسی دارند هزینه دسترسی میانگین دارد کاهش میابد بنابراین تاخیر این انتقالات کاهش می یابد و زمان پاسخ[9] بهتر می شود[11] در الگوریتم NNA یک حد آستانه ای که مقدار آن برابر ۵ بود برای انتقال قطعه های داده در نظر گرفته می شد یعنی اگر شمارنده مربوط به یک قطعه داده که توسط نودهای دیگر دسترسی می شد مساوی ۵ می شد این قطعه داده با استفاده از الگوریتم های مسیر یابی به یکی از نودهای همسایه منتقل می شد در حالیکه در الگوریتم RTNNA این مقدار آستانه نسبی می باشد بدین معنا که با توجه به مرحله دوم الگوریتم شمارنده ساده (Simple Algorithm Counter) تصمیم گیری درباره انتقال قطعه داده به این صورت انجام می شود که سطرهای یک بلاک در صورتی که شمارنده مربوط به آن بلاک در یک سایت بیشتر از سایتی باشد که بلاک هم اکنون در آن قرار دارد تصمیم گیری درباره انتقال قطعه داده با استفاده از الگوریتم های مسیر یابی انجام می شود که در این پروژه ما مانند الگوریتم NNA از الگوریتم link state استفاده کردیم برای اینکه مشخص کنیم که قطعه داده بایستی در کدام نود قرار گیرد که میانگین دسترسی نودهای دیگر به این قطعه داده کمترین مقدار را داشته باشد در این الگوریتم (RTNNA) بهبود قابل ملاحظه ای نسبت به الگوریتم NNA حاصل شد که نتایج آن در ادامه آورده شده است .

با توجه به شکل ۱ فرض کنید نود I، H، G وE به طور مکرر برای قطعه داده i که در نود A قرار دارد درخواست می فرستند با توجه به الگوریتم RTNNA هنگامی که شمارنده مربوط به یکی از این نودها بالاتر از بقیه نودها و نود مبدا باشد قطعه داده i به نود C منتقل می شود اگر این درخواستها بعد از انتقال قطعه داده ادامه پیدا کند قطعه داده به نود B منتقل می شود این روش ادامه پیدا می کند تا وقتی که قطعه داده به نود G منتقل شود . با قرار گرفتن این قطعه داده در نود G ، درخواستها از نودهای I، H، G و E با کمترین تاخیر و نه تاخیر کمینه جواب داده می شود در این مرحله داده در یک وضعیت پایدار قرار می گیرد بعد از این مرحله اگر یکی از نودهای H ،G وE بیشتر از دیگران درخواست قطعه داده بدهند این قطعه داده در این نود قرار می گیرد .



شکل ۱ : توپولوژی آزمایش

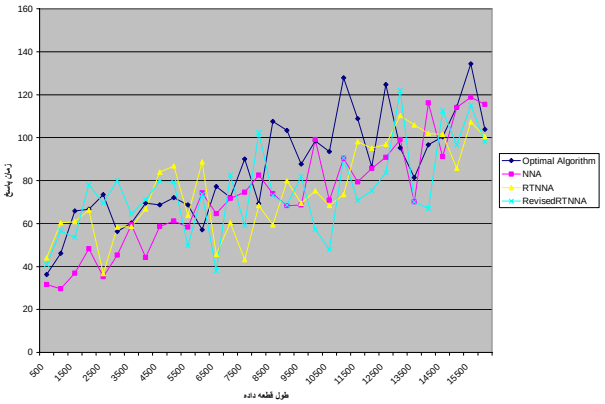
در آزمایشات انجام شده ما دو فاکتور در نظر می گیریم : تاخیر متوسط برای دریافت پاسخ از درخواست یک قطعه داده (زمان پاسخ [10]) و زمان سپری شده برای انتقال داده از یک نود به نود دیگر (زمان انتقال قطعه داده[11]) .

در الگوریتم Revise RTNNA تصمیم گیری برای انتقال قطعه داده از سایتی که در آن قرار دارد به سایت دیگر به این

[8] Routing
[9] Response Time
[10] Response Time
[11] Fragment Data Migration Time

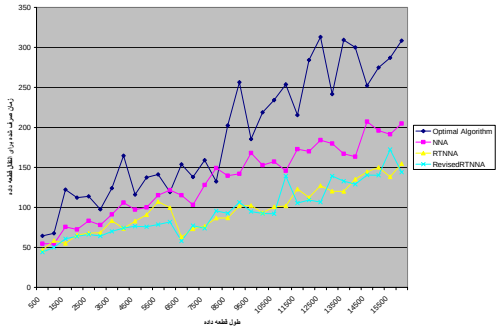
صورت انجام می شود که ابتدا مجموع شمارنده مربوط به یک قطعه داده که توسط تمامی سایتهای موجود در شبکه به آن دسترسی شده محاسبه می شود سپس این مجموع بر تعداد نودها منهای نودهایی که به این قطعه داده دسترسی پیدا نکرده اند تقسیم می شود از این معادله یک نتیجه حاصل می شود اگر این مقدار از مقدار شمارنده این قطعه داده مربوط به سایتی که در آن قرار دارد بیشتر باشد انتقال قطعه داده انجام می شود و داده یک مرحله با توجه به الگوریتم مسیریابی link state یک قدم منتقل می شود و در سایت دیگری قرار می گیرد در زیر دو الگوریتم RTNNA و Revise RTNNA با توجه به آزمایشهای انجام شده مقایسه شده است.

۳. محیط آزمایشها و نتایج شبیه‌سازی: همانطور که در شکل ۲ مشاهده می شود نتایج حاصله از مقایسه چهار روش optimal ، NNA ، RTNNA و Revise RTNNA نشان می دهد که برای فاکتور زمان پاسخ براساس طول قطعه داده الگوریتم NNA برای قطعه های داده با طول کمتر از ۶۰۰۰ بهتر از بقیه عمل می کند در بازه ۶۰۰۰ و ۹۵۰۰ نتایج حاصله از این چهار روش برای فاکتور پاسخ نزدیک هم می باشد ولی وقتی طول قطعه داده از ۹۵۰۰ بیشتر می شود الگوریتم RTNNA بهتر از سایر روش ها عمل می کند. همچنین الگوریتم Revised RTNNA برای برخی قطعه های داده خیلی خوب عمل می کند ولی برای برخی دیگر نتایج خوبی از این الگوریتم بدست نمی آید این بدین معنا نیست که این الگوریتم بدتر از بقیه عمل می کند. در یک مقایسه کلی می توان گفت که روش RTNNA بهترین عملکرد را برای فاکتور زمان پاسخ دارد بعد الگوریتم NNA نسبت به بقیه خوب عمل می کند و نوسان کمتری برای قطعه داده با طول مختلف دارد بعد الگوریتم Revise RTNNA برای برخی قطعه های داده خوب است و در نهایت الگوریتم optimal ضعیف ترین عملکرد را در بین این چهار روش دارد. در این مقایسه ما طول قطعه های داده را در بازه ۵۰۰ تا ۱۶۰۰۰ تغییر دادیم و به نتایج زیر رسیدیم :



شکل ۲ : تاثیر طول قطعه داده بر زمان پاسخ با استفاده از چهار روش optimal،NNA، RTNNA و Revise RTNNA

در شکل ۳ زمان صرف شده برای انتقال قطعه داده برای قطعه های داده با طول قطعه داده در بازه ۵۰۰ تا ۱۶۰۰۰ با استفاده از چهار روش optimal ، NNA ، RTNNA و Revise RTNNA نشان داده شده است همانطور که از این شکل بر می آید زمان لازم برای انتقال قطعه داده با استفاده از روش Revise RTNNA کمترین مقدار در بین این چهار روش را دارد بعد از این روش الگوریتم RTNNA زمان کمتری برای انتقال داده صرف می کند سپس الگوریتم NNA و در نهایت الگوریتم optimal بیشترین زمان را در بین این چهار روش برای انتقال قطعه داده صرف می کند. با دقت در شکل زیر می توان نتیجه گرفت که برای قطعه های داده با سایز کمتر از ۸۰۰۰ این اختلاف زیاد نمی باشد ولی با افزایش طول قطعه داده الگوریتم optimal بدترین عملکرد را دارد و اختلاف آن با سه روش دیگر به طور صعودی افزایش پیدا می کند و زمان زیادی را برای انتقال قطعه های داده صرف می کند همچنین الگوریتم NNA نسبت به الگوریتم های RTNNA و Revise RTNNA زمان زیادی را برای انتقال داده صرف می کند ولی این دو الگوریتم عملکرد مشابهی دارند و تقریباً برای انتقال قطعه های داده زمان یکسان صرف می کنند ولی الگوریتم Revise RTNNA در حالت کلی بهتر از همه عمل می کند و زمان لازم برای انتقال قطعه های داده با استفاده این الگوریتم مقدار کمینه می باشد.



شکل ۸ : زمان لازم برای انتقال قطعه داده براساس تغییرات طول قطعه داده با استفاده از چهار الگوریتم optimal ، NNA ، RTNNA و Revise RTNNA

۴. نتیجه‌گیری: در این مقاله یک الگوریتم تخصیص قطعه داده پویا به نام نزدیکترین همسایه با حد آستانه نسبی و نسخه اصلاح شده آن بررسی گردید. این الگوریتم‌ها بر پایه الگوریتم optimal می‌باشند ولی از استراتژی مختلفی برای انتقال داده به سایت مقصد استفاده می‌کنند. ما در آزمایشهای خود دو فاکتور درنظر گرفتیم که اولی میانگین تاخیر برای پاسخ دادن به درخواست قطعه داده بود و دومی زمان سپری شده برای انتقال قطعه‌های داده در شبکه بود. ما تاثیر پارامترهای مختلف را روی این فاکتورها بررسی کردیم. یافته‌های ما نشان می‌دهد که الگوریتم Revise RTNNA برای فاکتور زمان صرف شده برای انتقال قطعه داده بهتر از الگوریتم‌های دیگر عمل می‌کند البته در این فاکتور الگوریتم RTNNA هم بسیار خوب عمل می‌کند و بهبود بسیار خوبی با استفاده از این الگوریتم‌ها حاصل می‌شود ولی برای فاکتور زمان پاسخ این چهار روش نتایج تقریباً مشابه دارند با این حال الگوریتم RTNNA بهتر از بقیه می‌باشد.حد آستانه برای اندازه قطعه داده ۸۰۰۰ بایت می‌باشد. برای شبکه‌های بزرگتر ما می‌توانیم از الگوریتم RTNNA برای کاهش زمان انتقال داده استفاده کنیم.

در این مقاله ما این الگوریتم‌ها را در سیستمهای پایگاه داده توزیعی غیرتکراری بررسی کردیم . مطالعات بیشتری برای بررسی آنها در سیستمهای پایگاه داده توزیعی تکراری نیاز می‌باشد.

یک شبکه با تعداد نودهای بیشتری می‌تواند با استفاده از الگوریتم RTNNA بررسی گردد و با افزایش تعداد نودها این الگوریتم می‌تواند دوباره ارزیابی گردد.

۵. تقدیر و تشکر : در اینجا جا دارد از راهنمایی های موثر و دلسوزانه جناب آقای دکتر مسعود رهگذر استادیار گروه مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران تقدیر و تشکر نمائیم.

۶. منابع و مآخذ :

[1]L. C. John, A Generic Algorithm for Fragment Allocation in Distributed Database Systems, *ACM*, 1994

[2] Ahmad, I., K. Karlapalem, Y. K. Kwok and S. K. Evolutionary Algorithms for Allocating Data in Distributed Database Systems, *International Journal of Distributed and Parallel Databases*, 11: 5-32, The Netherlands, 2002.

[3] A. Brunstroml, S. T. Leutenegger and R. Simhal, Experimental Evaluation of Dynamic Data Allocation Strategies in a Distributed Database with changing Workloads, *ACM Transactions on Database Systems*, 1995

[4] A. G. Chin, Incremental Data Allocation and ReAllocation in Distributed Database Systems, *Journal of Database Management;* Jan-Mar 2001 ; 12, 1; ABI/INFORM Global pg. 35

[5] T. Ulus and M. Uysal, Heuristic Approach to Dynamic Data Allocation in Distributed Database Systems, *Pakistan Journal of Information and Technology 2 (3): 231-239*, 2003 , ISSN 1682-6027

[6] S. Voulgaris, M.V. Steen, A. Baggio, and G. Ballintjn, Transparent Data Relocation in Highly Available Distributed Systems. *Studia Informatica Universalis*. 2002

[7] Navathe, S.B., S. Ceri, G. Wiederhold and J. Dou, Vertical Partitioning Algorithms for Database Design, *ACM Transaction on Database Systems*, 1984 , 9: 680-710.

[8] P.M.G. Apers, “Data allocation in distributed database systems,” *ACM Transactions on Database Systems*, vol. 13, no. 3, pp. 263–304, 1988.

[9] Y. F. Huang and J. H. Chen, Fragment Allocation in Distributed Database Design *Journal of Information Science and Engineering 17*, 491-506, 2001

[10] I. O. Hababeh , A Method for Fragment Allocation Design in the Distributed Database Systems, *The Sixth Annual U.A.E. University Research Conference*, 2005

[11] R. Baseda, S. Tasharofi, M. Rahgozar, Near Neighborhood Allocation: A Novel Dynamic Data Allocation Algorithm in DDB, CSICC 2006.