ایجاد یک محیط برای شبیه سازی و سنجش مدلهای توزیع داده در پایگاه‌داده های توزیعی

کاوه پاشائی ، رضا باصدا
دانشگاه تهران- دانشکده فنی - گروه مهندسی برق و کامپیوتر

{K.Pashaei,R.Basseda}@ece.ut.ac.ir

چکیده : *محیطهای موجود برای شبیه سازی سیستمهای توزیع داده در پایگاه‌داده‌های توزیعی و کلا مکانیزمهای شبیه سازی شبکه های کامپیوتری ، همواره دارای پیچیدگی های بالا بوده و معمولا در محیط Linux انجام شده است. این محیطها معمولا به دلیل پیاده سازی لایه ای کلیه مکانیزمهای شبکه ، دارای پیچیدگی بالا بوده و به دلیل محیط پیاده سازی (که معمولا زبان C) میباشد ، و نیز نحوه پیاده سازی (بالا بودن حجم بالای کلاسها) و نداشتن مستندات لازم ، باعث بروز مشکلات بسیاری در امر توسعه پروتکل ها در قالب این نرم افزار ها شده است.*

توسعه یک ازار در یک زبان شی گرا این امکان را میدهد که بتوان ، سیستم را برای یک کاربرد خاص مانند توزیع داده در پایگاه‌داده‌های توزیعی یا کنترل همروندی و ... براحتی تغییر داده و نیاز به درتغییر زیاد در کل ساختار مجموعه نباشد.

همچنین ، تعداد متغیر های محدود در یک سیستم تخصصی ، به فهم سیستم کمک نموده و تغییرات در آن را آسان مینماید.

۱-مقدمه : سیستم توسعه یافته با عنوان یک محیط شبیه ساز الگوریتم های توزیع ، با توجه به نوع نگرش به مساله ، بطور خلاصه تنها به تعریف الگوریتم توزیع داده پرداخته و با توجه به ساختار الگوریتمهای توزیع داده ، تنها بخش تصمیم گیری در ایجاد انتقال داده را پیاده سازی نموده و بدلیل ساده شدن طراحی و تمرکز بر روی الگوریتم مسیر یابی ، نوع اطلاع رسانی به گره ها پیاده سازی نشده و پیاده سازی پروتکل بصورت استاندارد نیست

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 سیستم ، محیطی جهت آزمایش انواع متریکها پدید آمده است. با توجه به طراحی سنکرون در سیستم و وجود یک منبع دانش و یک جز هماهنگ کننده در مجموعه ، جریان اطلاعات در سیستم پیچیده نبوده و امکان تصمیم‌گیری بر اساس متریکهای سیستم بصورت دقیق وجود دارد.

بطور مثال ، از جمله مزایای سنکرون بودن سیستم ، امکان محاسبه دقیق تاخیر صف بندی در یک پیوند و یا تاخیر بسته های دریافتی در یک گره میباشد.

همانطور که قبلا گفته شد برنامه با توجه ساختار شی گرای خود از قابلیت توسعه برخودار بوده و اجزای مربوط به

سایر سرویسها و لایه ها در شبکه و پایگاه داده‌های توزیعی به راحتی به مجموعه قابل افزایش است.

برنامه بطور کلی دارای ۸ فایل در قالب زبان Java بوده و مشتمل بر ۷۰۰+۰۰۵۱ خط برنامه است که یکی از این کلاسها مربوط به واسط کاربر بوده و سایر کلاسها در هر فایل برای شبیه سازی ایجاد شده است. در ادامه این مستند به توصیف برنامه ایجاد شده پرداخته و هر قسمت را بطور جداگانه معرفی خواهیم نمود.

۲-کلاس DDBDataAlloc : در این کلاس اشیا اصلی برنامه نمونه گیری شده و میتوان گفت که این کلاس به نوعی مدیر برنامه محسوب میشود. در این کلاس از کلاس RoutingUI که کلاس مربوط به واسط کاربر میباشد نمونه گیری میگردد ولی کلاسهای مربوط به شبیه سازی شبکه از جمله coordinator و ... در این کلاس استفاده نشده است.

۳-کلاسRoutingUI : این کلاس برای پیاده سازی واسط کاربر ایجاد شده است. همانطور که در شکل ۱ مشاهده میشود در این کلاس از یک شی jTextField برای دریافت طول مدت زمان شبیه سازی از کاربر استفاده میشود. همچنین از تعدادی شی jRadioButton برای دریافت اطلاعات مربوط به نوع روش توزیع داده استفاده میگردد. لازم به ذکر است .

در یک بخش از این برنامه نیز میتوان فرکانس کلی ارسال بسته ها را در سناریوی داده شده به برنامه اعلام نمود. در نسخه فعلی برنامه سازمان و ساختار شبکه و نیز سناریوی برنامه در داخل برنامه به کلاسها ارائه میگردد ولی میتوان برنامه را طوری تغییر داد که بتواند سازمان شبکه و نیز سناریوی بسته های ارسالی را از کاربر دریافت نماید.

در بخش دیگری از این کلاس با استفاده از اشیا از نوع jTextArea اطلاعات مربوط به نتیجه شبیه سازی به کاربر ارائه میشود. اولین بخش از این گزارشات مربوط به وضعیت بسته های دریافتی در هر گره میباشد. این گزارش شامل تعداد بسته دریافتی میزان متوسط تاخیر در بسته های دریافتی میباشد.

در بخش بعدی از گزارش وضعیت مربوط به Link های موجود در شبکه ارائه شده است که نشانگر کارایی Link و نیز تعداد پالس زمانی است که Link بصورت بیکار بوده است. همچنین در ادامه این بخش وضعیت هر یک از جریان های داده ای بطور جداگانه بررسی شده است. همچنین در این بخش تعداد داده های قرار گرفته شده در هر گره را نیز نمایش میدهیم.

در نهایت یک جمعبندی از کل وضعیت شبیه سازی ارائه شده است. این جمعبندی شامل میانگین تاخیر در گره های شبکه تعداد بسته های Drop شده تعداد بسته های دریافت شده متوسط تاخیر در هر Link و میانگین کارایی در Link ها میباشد. دراین بخش میزان تاخیر در اثر انتقال داده و نیز تاخیر در دریافت پاسخ پرس و جو نیز نمایش داده می‌شود.

همانطور که مشاهده میشود با فعال شدن رخ داد مربوط به jButtonStartSim یک شی از کلاس coordinator ایجاد نموده و شبیه سازی را آغاز مینماید.

۳-کلاس pack : این کلاس اصلی ترین کلاس مربوط به انتقال داده میباشد و به نوعی نماینده بسته های منتقل شده حاوی Query و نیز حامل داده در شبکه میباشد.

در این کلاس داده های هدف، نوع بسته (شامل داده پایگاه داده، پرس و جو، پاسخ پرس و جو) و ... را نمایش می‌دهد.

علاوه بر موارد فوق در این بسته تعدادی متغیر صحیح وضعیت زمانی بسته را در برخی مواقع خاص مثلا داخل یک صف شدن یا ارسال شدن مشخص میسازد. همچنین توابع مربوط به محاسبه تاخیر صف بندی و تاخیر دریافت بسته نیز در این کلاس گنجانده شده اند. این مورد را میتوان به عنوان یک مزیت نسبت به شبیه ساز DBSim معرفی نمود. در DBSim ما مجبور بودیم که فایل trace ایجاد شده را توسط یک برنامه جانبی دیگر تفسیر نموده و نتایج را استخراج نمود ولی در اینجا برنامه قدرت محاسبه متریک شبکه را نیز دارد.

در این کلاس با استفاده از TimeStamp های موجود در کلاس، میزان تاخیر یک بسته را در هر مرحله از پیمایش محاسبه نمائیم.

۳-کلاس MyNode و کلاسهای بکار رفته در آن : این کلاس برای ایفای نقش یک گره در شبکه ایجاد شده است . در این کلاس تعداد بسته های دریافتی در هر گره محاسبه شده ونیز تاخیر هر بسته از کلاس pack دریافت شده و میزان متوسط تاخیر بدست می آید.

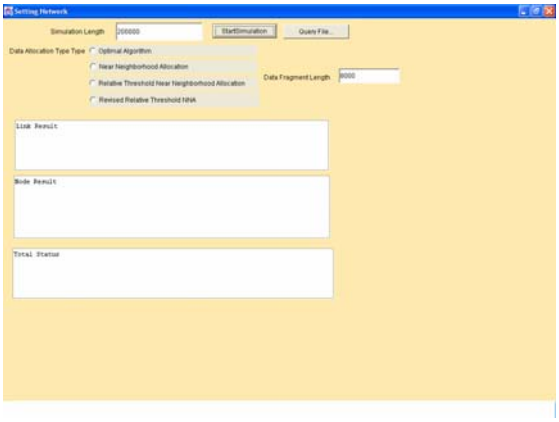
این کلاس یک کلاس queue دارد که نقش صف خروجی را برای هر Link ایفا میکند. همچنین یک کلاس rtTable در این کلاس وجود دارد که نقش جدول مسیر یابی را ایفا میکند.

در این کلاس ما ویژگی های مربوط به یک گره را در سیستم پیاده سازی نموده ایم . در این کلاس چند زیر کلاس وجود و اطلاعات را ایجاد مینماید.

۳,۱-کلاس Queue : در این کلاس یک صف برای هر Link پیاده سازی میگردد. این کلاس صرفا یک صف ساده است ولی میتوان با توسعه کد قابلیت توزیع بسته بصورت عادلانه را ایجاد نمود. این کلاس میتواند جهت پیاده سازی سیستم های صف بندی بکار رود.

۳,۲-کلاس rtTable : در این کلاس فرآیند مسیر یابی در سیستم قرار داده شده و با استفاده از متد hint اطلاعات مربوط به وضعیت شبکه را دریافت نموده و بر اساس آن مسیر یابی را انجام میدهد.

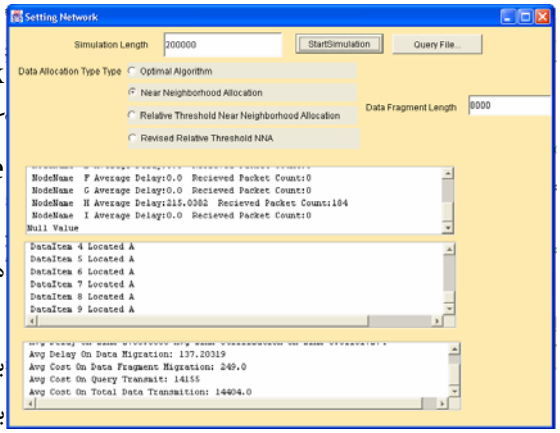
در این کلاس میتوان انواع متدهای مسیریابی را پیاده سازی نموده و متریکهای مسیریابی را در هر نوع مشخص نمود.



شکل 1 واسط کاربر برای درج پارامترهای شبیه سازی

با توجه به اینکه وضعیت شبکه با استفاده از کلاس coordinator قابل بازیابی است میتوان گفت که این سیستم احتیاجی به پروتکلهای انتقال پیام برای توصیف شبکه ندارد.

در این کلاس در هنگام ایجاد هیپگونه مسیر سازی انجام نمیشود زیرا مسیر ها در هنگام ساخته شدن این کلاس (یعنی هنگام نمونه سازی کلاس container) مشخص نشده اند.



شکل 2 نتایج یک نمونه شبیه سازی و آزمایش

۳,۳-کلاس PacketGenerator : در این کلاس عملیات تولید بسته پرس و جو برا ساس مبنای فرکانسی داده شده توط کاربر انجام میشود. در این کلاس به ازای یک flow بسته ها ایجاد شده و توسط گره مبدا Get میشوند.

این کلاس وظیفه زماندهی بسته تولید شده را نیز دارد. در الگوریتمهای کنترل کننده بار پایگاه داده توزیعی این کلاس جهت تست در صورت refuse شدن درخواست تولید بسته قطع شده و تا زمان بیدار شدن مجدد متوقف میگردد.

۴-کلاس MyLink: در این کلاس وضعیت یک Link در شبکه مشخص شده و فرآیند انتقال بسته با توجه به پهنای باند و تاخیر Link انجام میپذیرد. در این کلاس فرآیند محاسبه کارایی یک Link انجام شده و تعداد پالسهایی که Link بصورت خالی میباشد را محاسبه مینماید. این حالت میتواند تسهیم مطلوب منابع شبکه را توسط الگوریتم مسیر یابی مشخص میگردد.

در این کلاس فراخوانی مربوط به دریافت بسته توسط کلاس گره انجام میشود. این کلاس بصورت خودکار هنگام رخداد مناسب بسته را منتقل مینماید.

لازم به ذکر است در حالت فعلی نمیتوان بصورت متناوب Link را از بسته های متوالی پر کرد در هر حال میتوان گفت که در صورت تایید صف یک Link میتوان Link را بهتر Utillize کرد.

۵-کلاس coordinator : در این کلاس عملیات هماهنگی و مدیریت زمان و اشیا انجام میشود.

در ایت کلاس چیدمان شبکه ارائه شده و یک سناریو برای ارسال بسته داده میشود. در این کلاس زمانبندی ارسال با یک تفاوتهایی از کاربر پرسیده میشود.

این کلاس جمع بندی نهایی شبیه سازی را ارائه نموده و فاکتورهای لازم را از تمامی اشیا خود استعلام مینماید.

۶-کلاسDataAlloc : این کلاسی است که در آن مکانیزمهای انتقال داده و نوع تصمیم گیری فراهم شده است. شمارنده های ارجاع به داده ها و ... قرار دارد.

در این کلاس دو روش مجزا برای تخصیص Fragment داده به سیستم در نظر گرفته شده است. که شرح این دو روش در مقاله [5] آمده است. در حالتی که پارامتر AllocationType برابر N باشد تخصیص Fragment بصورت ارسال به درخواست کننده می باشد. در غیر اینصورت Fragment به گره بعدی ارسال می گردد.

۷- تکیمل سیستم : با توجه به ساختار موجود در برنامه قابلیت ارتقا سیستم وجود داشته و مقرر گردیده تا با همکاری یک از دانشجویان درس پایگاه داده پیشرفته سیستم ایجاد شده تکمیل گردد.

۸- نحوه راه اندازی سیستم : با توجه به ساختار موجود در برنامه پس از نصب نرم افزار jBuilder بایستی فایل DDBFragment توسط این برنامه باز شده و سپس سیستم اجرا گردد. بایستی در اجرای شبیه سازی طول شبیه سازی و نیز طول هر Fragment داده برای سیستم از طریق واسط کاربر مشخص شود.

۹- 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.

۱۰- 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.