مجتمع سازی داده‌ها در محیط p2p

حسن شجاعی مند

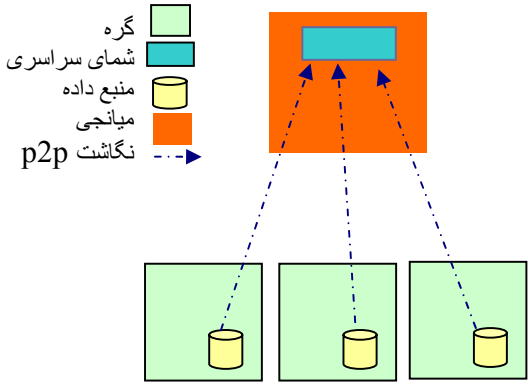
مجتمع‌سازی داده عبارت است از ترکیب داده‌های موجود در منابع‌داده مختلف به‌گونه‌ای که کاربر بتواند با استفاده از یک دید واحد به داده‌ها دسترسی پیدا کند. فرض کنید که تعدادی منبع‌داده یا به صورتی آشناتر Database داشته باشیم. هر کدام از این منابع‌داده یکسری داده‌هایی در خود ذخیره کرده‌اند که همگی به یک دامنه خاص مربوط می شوند. به عنوان مثال دامنه بیماران را در نظر بگیرید. برای این دامنه، بانک‌های اطلاعاتی که در بیمارستان‌ها، درمانگاه‌ها و مطب‌های شخصی وجود دارد به عنوان منابع داده موجود در نظر گرفته می شوند. اجازه بدهید چند بیمارستان و مطب شخصی را به عنوان نمونه انتخاب کنیم و شمای بانک اطلاعاتی آنها را با جزئیات بیشتری بررسی کنیم.

بانک اطلاعاتی بیمارستان قائم دارای دو جدول patient و treatment می باشد. Patient حاوی اطلاعات بیماران می باشد(hid شناسه،ni شماره بیمه،name نام بیمار،sex جنسیت بیمار و gp دکتر خانوادگی بیمار می باشد).treatment اطلاعات مربوط به معاینات انجام شده بر روی بیمار را نگهداری می کند(tid شناسه،hid شناسه بیمار،date تاریخ،desc توضیحاتی در خصوص معاینه و درمان انجام شده و consultant پزشکی است که بیمار را معاینه کرده است).

PS1	patient(hid,ni,name,sex,gp) treatment(tid,hid,date,desc,consultant)
PS2	Patient(ni,fName,lName,sex,address) Event(ni,date,desc)
PS3	p (id,name,doctor) t (pid,ni,date,desc)

شکل۱: شمای بانک‌های اطلاعاتی.PS1 شمای بانک اطلاعاتی بیمارستان قائم، PS2 شمای بانک اطلاعاتی مطب دکتر مختاری و PS3 شمای بانک اطلاعاتی درمانگاه بهگر می باشد.

مجتمع سازی داده به ما این امکان را می دهد که به اطلاعات تمام این منابع داده از طریق یک سیستم واحد دسترسی پیدا کنیم. دو روش کلی برای اینکار وجود دارد. که این دو روش را تحت عنوان‌های مجتمع سازی داده‌ها با استفاده از میانجی و مجتمع سازی داده‌ها در محیط p2p، در ادامه مورد بررسی قرار می دهیم.



شکل۳: مجتمع سازی داده‌ها با استفاده از میانجی

۱- مجتمع سازی داده‌ها با استفاده از میانجی

این روش در [1] آمده است و ساختار کلی آن در شکل۲ نشان داده شده است. در کنار منابع داده موجود، یک میانجی به سیستم اضافه می شود. شمایی برای آن تعریف می شود که بتواند دربرگیرنده کلیه اطلاعات موجود در دامنه مورد نظر باشد. به این شما، شمای سراسری می گوییم. سپس با استفاده از یکسری قواعد به نام قواعد نگاشت، ارتباط بین شمای منابع داده با شمای سراسری را برقرار می کنیم. شمای سراسری با استفاده از

قواعد نگاشت می داند که چه اطلاعاتی را از کجا می تواند بدست آورد. در ادامه مثالی که در مقدمه شروع کردیم، فرض کنید که شمای سراسری در نظر گرفته شده برای میانجی به صورت زیر باشد:

AllPatient(ni,name,sex,address,gp)
AllTreatment(ni,description,date)

چند روش مختلف برای تعریف قواعد نگاشت بین دو شما در مقالات پیشنهاد شده است[1,7] که ایده کلی به این صورت است که جداول یک شما را بر حسب یک ویو بر روی جداول شمای دیگر بنویسیم. این ویو ارتباط جداول دو شما را مشخص خواهد کرد. نحوه تعریف قواعد نگاشت را سعی می کنیم تا با دنبال کردن مثالی که در پیش گرفته‌ایم، بهتر متوجه شویم:

$$PS1.patient(\text{ni, name, sex, gp}) \subseteq \text{AllPatient(ni, name, sex, gp)}$$
$$\text{PS2.patient(ni, fName+''+lName, sex, address)} \subseteq \text{AllPatient(ni, name, sex, address)}$$
$$\text{PS3.p(id, name, gp = doctor), PS3.t(-, id, ni,-,-)} \subseteq \text{AllPatient(ni, name, gp)}$$

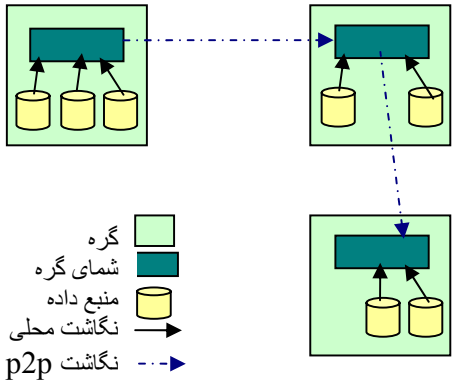
حال اگر یک پرس‌وجو به شمای سراسری اعمال شود، با توجه به قواعد نگاشت تعریف شده فوق می داند که باید اطلاعات را چگونه و از کجا بدست آورد. پرس‌وجو را با استفاده از همین قواعد به پرس‌وجوهایی بر روی شمای منابع داده تبدیل می کند و به آنها ارسال می کند تا جواب برگشتی را به عنوان جواب پرس‌وجو در اختیار کاربر قرار دهد.(در قسمت چپ نگاشت سوم الحاق بین دو جدول نشان داده شده است بر روی صفت مشترک id)

در این روش همانطور که با هم دیدیم، میانجی همه کاره است. اگر بخواهیم جواب یک پرس‌وجو تمام منابع داده را شامل شود باید آنرا به شمای سراسری میانجی اعمال کنیم. منابع داده با یکدیگر رابطه‌ای ندارند و نمی توانند از اطلاعات یکدیگر بهره‌مند شوند. از طرف دیگر وجود میانجی خود می تواند به عنوان یک تنگنا برای سیستم به حساب آید و چون ارتباط با تمام منابع داده باید از طریق میانجی انجام شود، بنابراین آزادانه نمی توان تعداد منابع داده را افزایش داد. از طرف دیگر در این روش باید در ابتدا یک شمای سراسری برای سیستم تعریف شود و تعریف شمای سراسری در ابتدا مشکل می باشد و به مرور زمان با اضافه شدن منابع داده جدید ممکن است تغییر کند و این تغییرات باید در قواعد نگاشت تعریف شده به‌گونه‌ای منعکس شوند. انعکاس تغییرات در قواعد نگاشت کار پرهزینه‌ای می باشد. تمام این موارد باعث می شود تا به دنبال راه بهتری برای مجتمع سازی باشیم.

اگر در مشکلات عنوان شده فوق، دقت کنید می بینید که همه این مشکلات به نوعی به میانجی برمی گردد. میانجی‌ای که از خود هیچ ندارد و در این سیستم ارزشی بیش از منابع‌داده یافته است. توجه به همین نکته می تواند آغازی برای رسیدن به روش دوم باشد.

۲- مجتمع سازی داده‌ها در محیط p2p

چارچوب این روش در [5] معرفی شده است. ساختار کلی آنرا در شکل۳ می توانید ببینید. در این روش همانطور که در شکل دیده می شود، میانجی وجود ندارد. در هر گره شمایی اضافه شه است به‌نام شمای گره. این شما حاوی اطلاعاتی است که گره می تواند از منابع داده محلی خود یا دیگر گره‌ها بدست آورد. هر گره بسته به توانایی‌اش حاوی یک یا چند منبع داده می باشد. شمای گره، حاوی کلیه اطلاعاتی است که یک گره در منابع داده محلی خود دارد یا می تواند از گره‌های دیگر درخواست کند. در این ساختار



شکل۳: مجتمع سازی داده‌ها در محیط p2p

یک گره در صورتی امکان حضور می یابد که حداقل دارای یک منبع‌داده مفید برای کل سیستم باشد. هر گـره از طریـق یکسری قواعد نگاشت که آنرا قواعد نگاشت p2p می نامیم می تواند به اطلاعات بقیه گره‌ها دست یابد. اجازه بدهیـد تـا مثالی را که در ابتدای مقاله شروع کرده‌ایم با این روش پیاده سازی کنیم تا جزئیات کار مشخص شـود. در ایـن سیسـتم نیازی به وجود میانجی و بنابراین شمای سراسری نداریم. سه شمای تعریف شده در شکل۱ می تواننـد از طریـق قواعـد نگاشت زیر با هم ارتباط برقرار کنند:

$$1:\ PS2.patient(ni,\ fName+''+lName, sex) \subseteq PS1.patient(ni, name, sex, 'Mokhtari')$$

$$2:\ PS1.patient(ni, name, gp) \subseteq PS3.p(id, name, gp = doctor),\ PS3.t(-, id, ni, -, -)$$

با استفاده از این دو قاعده نگاشت، تمام گره‌ها به اطلاعات یکدیگر می توانند دسترسی پیدا کنند. به عنوان مثال اگر یک پرس‌وجو بر روی شمای PS3 اعمال شود، این گره ابتدا جواب‌های پرس‌وجو را از منابع داده خود بدست می آورد. سپس با استفاده از قاعده نگاشت دوم آنرا به پرس‌وجویی بر روی شمای PS1 تبدیل می کند و آنرا برای PS1 ارسال می کنـد. PS1 جواب پرس‌وجو را از منابع داده خود بدست آورده و برای PS3 می فرستد، و سپس با استفاده از قاعده نگاشت اول پرس‌وجو را به پرس‌وجویی بر روی شمای PS2 تبدیل کرده و این پرس‌وجو را به PS2 می فرستد و....

از مزایای این روش می توان به موارد زیر اشاره کرد:

- به وجود شمای سراسری نیازی نداریم و بنابراین مشکلات مربوط به آن نیز که در قسمت۱ توضیح داده شد در این سیستم وجود ندارد.
- گره‌های جدید می توانند براحتی وارد سیستم شوند. تنها کافیست تا با نزدیک‌تـرین گـره بـه خـود نگاشـتی را برقرار کنند. در سیستم بررسی شده قبلی، میانجی به عنوان یک تنگنا، اجـازه گسـترش سیسـتم را از مـا مـی گرفت.
- در این سیستم، از هر گره می توان به اطلاعات تمام گره‌ها دسترسی پیدا کرد. در صـورتیکه در سیسـتم قبلـی، تنها از طریق میانجی می شد به اطلاعات موجود در منابع داده دست یافت.

مراجع

[1] M. Lenzerini. Data integration: A theoretical perspective. In Proc. of the 21st ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2002), pages 233-246, 2002.

[2] A. Y. Halevy. Answering queries using views: A survey.Very Large Database J., 10(4):270–294, 2001.

[3] O. Duschka. Query Planning and Optimization in Information Integration. PhD thesis, Stanford University, 1997.

[4] S. Gribble, A. Halevy, Z. Ives, M. Rodrig, and D. Suciu. What can databases do for peer-to-peer? In Proc. of the 4th Int. Workshop on the Web and Databases (WebDB 2001), 2001.

[5] D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Logical foundations of peer-to-peer data integration. 2004. To appear.

[6] P. McBrien and A. Poulovassilis. Defining peer-to-peer data integration using both as view rules. In *Databases, Information Systems, and Peer-to-Peer Computing*, pages 91–107. Springer, LNCS 2944, 2004

[7] P. McBrien and A. Poulovassilis. *Data Integration by bi-directional schema transformation rules*. In Proc. ICDE'03. ICDE, March 2003.

[8] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, Z. Xu. *Peer-to-Peer Computing*. Internal HP Report,2002.