

پروتکل های کنترل همروندی:

رویکردهای زمانبندی تراکنش ها:

- خوش بینانه
- بدبینانه

خوش بینانه: فرض می شود که اجرای همروند تراکنش ها شرایط صحت همروندی را حفظ کند در نتیجه اجازه می دهیم که تراکنش ها اجرا شوند و در آخر قبل از نوشتن روی رسانه و انعکاس تغییرات صحت عملکرد آنها را بررسی می کنیم.

بدبینانه: فقط در صورتی که صحت و جامعیت بانک اطلاعات را حفظ کند به آنها اجازه اجرا بدهیم.

پروتکل های کنترل همروندی:

زمانبند درمقابل با هر عملگر، یکی از اعمال زیر را انجام می دهد:

- اجرای عملگر (execute)
- به تاخیر انداختن اجرای عملگر (delay)
- نپذیرفتن عملگر (reject)

انواع زمانبند ها:

- زمانبند محافظه کار (conservative)
- زمانبند بی پروا (aggressive)

انواع زمانبندها:

زمانبند محافظه کار: اجرای دستورات تراکنش را اگر لازم باشد به تعویق می اندازد و محتاطانه اجرای دستورات را دنبال می کند تا تراکنش ها ساقط نشوند.

زمانبند بی پروا: هدف پرهیز از تاخیر در اجرای دستورات است، پس دستورات را فوراً اجرا می کند.

انواع پروتکل های کنترل همروندی:

- پروتکل های مبتنی بر قفل (lock-based)
- پروتکل های مبتنی بر گراف (graph-based)
- پروتکل های مبتنی بر مهر زمانی (timestamp-based)

پروتکل های مبتنی بر قفل (lock-based)

این روش بر اساس تخصیص داده ها به تراکنش هاست، هرگاه تراکنشی بخواهد برای خواندن یا نوشتن به داده ای دسترسی داشته باشد، ابتدا درخواست قفل مناسب با آن دستور را به واحدی به نام مدیر قفل می دهد.

انواع قفل ها:

قفل باینری: در این قفل داده یا قفل است یا باز. اشتراک داده ها وجود ندارد و درخواست تراکنش ها فقط در صورت باز بودن قفل داده مربوطه اجابت می شود.

قفل اشتراکی-انحصاری (shared-exclusive): به منظور بالا بردن سطح همروندی تراکنش ها و امکان به اشتراک گذاشتن داده ها در حالات ممکن، قفلها به دو نوع اشتراکی (S) و انحصاری (X) تقسیم می شوند.

قفل اشتراکی-انحصاری:

قفل اشتراکی: برای خواندن داده استفاده می شود. پس در صورت نیاز به اجرای دستور خواندن باید درخواست قفل اشتراکی بدهیم و مدیر قفل به سایر تراکنش ها نیز اجازه قفل اشتراکی و خواندن داده را می دهد.

قفل انحصاری: برای نوشتن داده استفاده می شود. داده مربوطه را به طور انحصاری قفل و آن را در اختیار تراکنش قرار می دهد. هیچ تراکنش دیگری نمی تواند به آن دسترسی داشته باشد.

قفل اشتراکی-انحصاری:

- درخواست قفل به مدیر قفل داده می شود تا در صورت سازگاری اجابت شود.
- اگر درخواست اجابت نشود تراکنش به حالت انتظار می رود.
- چندین تراکنش می توانند به طور همزمان یک داده از نوع s را قفل کرده و همزمان بخوانند. اما اگر قفل آن از نوع x باشد هیچ تراکنش دیگری نمی تواند روی آن قفل بگذارد.
- درخواست قفل s برای تراکنش i ، $si(Q)$ و درخواست قفل x برای تراکنش i ، $xi(Q)$ و درخواست آزاد شدن قفل $ui(Q)$
- قفل گذاری داده ها می تواند با دانه بندی های مختلف انجام شود.
- برای پیاده سازی واحد مدیر قفل از جدول قفل استفاده می شود.

افزایش و کاهش درجه قفل:

یک تراکنش باید در صورت لزوم قفل خود را تبدیل کند.

افزایش درجه قفل: اگر تراکنشی قبلاً قفل $s(Q)$ دارد و بعداً می خواهد عمل $w(Q)$ را انجام دهد باید درخواست $x(Q)$ را بدهد.

کاهش درجه قفل: اگر قفل $x(Q)$ برای خواندن بعدی به $s(Q)$ تبدیل شود تا دیگران هم بتوانند از این داده استفاده کنند. از مزایای آن افزایش درجه همروندی تراکنش هاست.

بن بست (deadlock) و قحطی (starvation) :

اگر یک انتظار چرخشی بین دو تراکنش به وجود آید به طوری که هیچ یک نتواند به کار خود ادامه دهد. این انتظار چرخشی منجر به وقوع بن بست می شود.

یک راه حل بن بست این است که یکی از این تراکنش ها را ساقط کنیم تا قفل هایش آزاد شود و دیگری بتواند با گرفتن آن قفل به کارش ادامه دهد. اما این راه حل قحطی را به وجود می آورد.