مقدمه

رشد وبالیدگی تکنولوژی سیستم مدیریت پایگاه داده (SBMS) با توسعه های قابل ملاحظه درفن آوری های پردازی موازی وتوزیعی مصادف ومقارن شده است. هدف نهایی آن ظهور وپدیداری سیستم های مدیریت پایگاه داده توزیعی وسیستم های مدیریت پایگاه داده موازی می باشند. این سیستم ها شروع به ایجاد به ابزارهای مدیریت داده قوی وهمه گیری برای کاربردهای داده ای پیچیده کرده اند تراکم وتجمع ایستگاه های کاری درمحیطی توزیع شده، توزیع کارمناسب تروکاراتری را باعث می شود، به صورتیکه برنامه های کاربردی، درایستگاه های کاری اجرا می شدند که درمرورهای کاربردی نامیده می شوند، درحالیکه کارهای پایگاه داده با کامپیوترهای اختصاصی که سرورهای پایگاه داده نامیده می شوند، اداره می شود. این امر، منجربه گرایش وتمایل کنونی به معماری وطراحی سیستم توزیعی می شود، که سایت ها وبخش های آن برای سرورهای خاص به گونه ای سازماندهی می شوند که بیش ازیک کامپیوترچند منظوره می باشد.

یک کامپیوترموازی مترجم، کامپیوتری که ازچند پردازنده با اتصال موازی استفاده می کند، خودش یک سیستم توزیعی است که ازتعدادی گره (پردازشگرها وحافظه ها) تشکیل شده است که با یک شبکه سریع درون یک کابینت اتصال می یابد. فن آوری پایگاه داده توزیعی می تواند به طورطبیعی برای اجرای سیستم های پایگاه داده موازی، بسط یافته وارتقا یابد یعنی سیستم های پایگاه داده درکامپیوترهای

موازی. سیستم های پایگاه داده موازی، درمدیریت داده، توازی گری (ترادف) را به جهت تحویل واجرای با دسترسی بالا وعملکرد بالای سرورهای پایگاه داده، درهزینه ای بسیارپائین ترازکامپیوترهای بزرگ (یک کامپیوترسطح بالا که برای امورمحاسباتی بسیارسنگین طراحی شده است) معادل خود، اعمال می کنند.

دراین مقاله، ما بررسی اجمالی ازفن آوری های DBMS توزیعی و DBMS موازی، با برجسته کردن ویژگی های مناسب هرکدام، داشته وتشابهات میان آنها را نیزنشان می دهیم. این بررسی، می تواند به نقش مناسب ومکمل آن دودرمدیریت داده، کمک کند.

مفاهیم (اصول) اساسی:

یک پایگاه داده توزیعی (DDB) مجموعه ای ازپایگاه داده های چندگانه ووابسته منطقی است که دریک شبکه کامپیوتری توزیع شده است.

یک سیستم مدیریت پایگاه داده توزیعی (DBMS توزیعی) به عنوان سیستم نرم افزارتعریف می شود که اجازه مدیریت پایگاه توزیعی را داده وتوزیع را برای کاربران « ناپیدا » (درمورد کامپیوتر، صنعتی برای یک وسیله، تابع یا بخشی ازبرنامه است که آنقدربه ملایمت ونرمی کارمی کند که برای کاربرقابل رویت نیست) می کند. این تعاریف به دواصل طراحی شناخته شده، اشاره می کنند. اولی آن است که سیستم، حاوی مجموعه ای (احتمالا خالی) ازبخشهای پرس وجو (مجموعه دستورالعمل های خاصی است که برای استخراج داده ازپایگاه داده ها وبه کارگیری

آنها) ومجموعه ای غیرتهی ازبخش هـای (سـایت هـا) داده مـی باشـند. بخـش داده ظرفیت ذخیره داده ها را دارند درحالیکه بخشهای پرس وجو این ظرفیت وتوانـایی را ندارند. یعنی بخشهای پـرس و جـو فقـط وال هـای رابـط کـاربررا بـه جهـت تسـهیل دسترسی به داده ها دربخشهای داده، اجرا می کنند. دومین اصل آن اسـت کـه بـرای هرسایت (بخش) توسط یک شبکه کامپیوتری که بیش از یک پیکربندی چند منظـوره است، به هم متصل می شوند. نکته مهم دراینجا تاکید بـرروی اتصـال ضـعیف میـان پردازشگرها است که سیستم های عامل خود را داشته به مستقل عمل می کنند. پایگاه داده به صورت فیزیکی درسایت هـای داده بـا قطعـه قطعـه سـاختن (چنـد پـارگی) وتکرارسازی داده ها، توزیع می شوند. با دادن یک طرح نسبی پایگاه داده، قطعه ها به هررابطه دربخشهای عمودی یا افقـی، تقسـیم مـی شـوند. قطعـه قطعـه شـدن (چنـد پارگی= بخش بخش کردن وتقسیم کردن درستون ها) افقـی یـک رابطـه توسـط یـک عملیات گزینشی انجام می گیرد کـه هرچنـد تـایی (درجـدول، پایگـاه داده رابطـه ای، مجموعه ای ازمقادیر مرتبط است که هریـک ازآنهـا بـه سـتون مربـوط مـی شـوند) ازرابطه را دریک پارتیشن مختلف برمبنای یک پیش بینی قطعه قطعه ای قرارمی دهد. (مثلا یک رابطه (کارمند) ممکن است طبـق مکـان (کارمنـدان) تقسـیم بنـدی وبخـش بخش گردد.

قطعه قطعه سازی عمودی، یک رابطه را برتعدادی ازفراگمنت ها (بخش ها) برحسـب صفات آنها، تقسیم می کند. (به عنوان مثال، رابطه « کارمند» می توانـد بـه قسـمتهای

همچون اطلاعات FMP-namber (شماره کارمند)، EMP-name (نام کارمند) و EMP-address (آدرس کارمند) دریک بخش (فراگمنت)، اطلاعات و EMP-number– EMP-salary (حقوق کارمند) ومدیرکارمند دربخش دیگر، قطعه قطعه شده وتقسیم گردد، قطعه قطعه سازی، امرمطلوبی است زیرا این کارقرارگیری داده های را درمجاورت نزدیک به مکان استفاده آنها، میسرمی سازد، ازاینرو هزینه مخابره وارسال به صورت بالقوه ای کاهش یافته واین امراندازه رابطه هایی را که دربخشهای پرس و جوی کاربرهستند، را می کاهد.

برمبنای الگوهای دسترسی کاربران، هریک ازفراگمن ها می توانند تکرارنیزشوند. این یک مزیت برتراست که داده های مشابه (همان داده ها) ازبرنامه های کاربردی که درتعدادی ازسایت ها (بخش ها) اجرا می شوند، دردسترسی قرارگیرند. دراین مورد، کپی کردن داده ها درتعدادی ازبخش ها می تواند نسبت به انتقال وحرکت دادن مداوم آنها میان سایت ها (بخش ها) ازنظرهزینه به صرفه ترومناسب ترباشند. هنگامی که فرضیه های طراحی بالا دریک DBMS توزیعی انجام می پذیرند، می توان به یک سیستم پایگاه داده موازی دست یافت. تفاوت های میان یک DBMS موازی ویک DBMS توزیعی، تا اندازه ای مجهول بوده وروشن نیستند. به ویژه، معماری های DBMS موازی با هیچ– اشتراک گذاری ای، که ما درزیردرمورد آنها بحث می کنیم، کاملا شبیه به همان مسئله سیستم های توزیعی با اتصال ضعیف هستند. DBMS های موازی ازمعماری های (طراحی های) کامپیوترچند پردازشگراخیربه جهت

ساختن سرورهای پایگاه داده با عملکرد ودسترسی بالا درهزینه بسیارپایین ترازکامپیوترهای بزرگ نظیرخود، بهره می برند.

یک DBMS موازی می تواند به عنوان DBMS ای تعریف شود که بریک کامپیوترچند پردازشگراجرا می شود. این سیستم شامل گزینه های بسیاری است ازپورتینگ (درسخت افزار، محلی برای وارد وخارج کردن داده ها ازیک دستگاه اطلاق می شود) ساده یک DBMS موجود، که ممکن است تنها فقط روال های رابط سیستم عامل را بازنویسی کند، گرفته تا یک ترکیب پیچیده ازپردازش موازی وعملیات های سیستم پایگاه داده دریک معماری سخت افزار/ نرم افزارجدید. مطابق همیشه، ما رابطه میان احتمال (برای چندین پلات فرم) وبهره برداری را سبک سنگین می کنیم. رویه ورویکردی پیچیده ودرسطح بالا، بهره برداری کامل ازفرصتهای ارائه شده توسط یک مولتی پروستور (چند پردازنده) درهزینه احتمالی، میسرمی سازد.

بنابراین، راه حل، استفاده ازموازی گر (ترادف) درابعاد بزرگ برای اهمیت دادن وبرجسته کردن قدرت خام اجزا و مولفه های تکی با متراکم کردن وجمع کردن آنها دریک سیستم کامل به همراه نرم افزارپایگاه داده موازی مناسب، می باشد. استفاده ازقطعات سخت افزاری استاندارد برای بهره برداری ازارتقا ها وپیشروی های تکنولوژیکی که مداوم با کمترین تاخیر، ضروری وحیاتی است. ازاینرو، نرم افزارپایگاه داده می تواند ازسه شکل موازی گری که ذاتا درحجم کاری برنامه های کاربردی با داده های فشرده وجود دارد، بهره برداری واستفاده کند. موازی گری

درپرس وجو، (بخش جستجو)، اجرای موازی چندین جستجو (پرس وجو) که توسط چندین تبادل همزمان ایجاد می شوند را میسرمی کند.

موازی گری درون جستجویی، اجرای موازی چند عملیات مستقل (مثلا عملیات گزینشی) را درهمان جستجو (پرس وجو) ممکن می کند. هردو مورد موازی گری درجستجو (inter- query) وموازی گری درون جستجویی (intra-quary) می توانند با استفاده ازافزار (جداسازی وتقسیم بندی) (پارتیشن بندی) داده ها که شبیه به قطعه قطعه سازی افقی است، حاصل شوند. درنهایت با موازی گری درون جستجویی، اضافه افرازداده، اعمال گردد. وضعیت عملیاتی زبان های پایگاه داده (مثلا SQL) فرصتهای زیادی برای موازی گری درون جستجویی، فراهم می آورد.

تعدادی ازمشخصه های شناساننده تکنولوژی DBMS توزیعی موازی:

۱- پایگاه داده موازی/ توزیعی، یک پایگاه داده است که مجموعه ای ازفایلهایی که می توانند به صورت منفرد درهرگروه یک شبکه کامپیوتری ذخیره گردند. این تمایزی میان یک DDB و مجموعه ای ازفایل هایی است که توسط یک سیستم فایل توزیعی مدیریت می شوند. برای تشکیل یک DDB ، داده های توزیعی باید ازنظرمنطقی رابطه داشته باشند، که این رابطه طبق برخی فرمالیسم های (آداب) ساختاری، تعریف می شوند. یعنی مدل رابطه ای و دستیابی به داده ها باید درسطح بالایی ازطریق یک رابطه مشترک باشد.

۲- سیستم عاملیت کامل یک DBMS را دارد. هماهنگونه که دربالا ذکرشد این نه یک سیستم فایل توزیعی، ونه یک سیستم پردازش تبادلی. پردازش تبادلی تنها یکی ازکارکردها وتوابع است که توسط چنین سیستمی ایجاد می شود واین سیستم، همچنین، کارهایی مانند پردازش جستجو، سازمان داده ها ودیگرکارهایی را که سیستم های پردازش تبادلی ضرورتا انجام نمی دهند، را پشتیبانی می کند.

۳- توزیع (شامل قطعه قطعه کردن وتکرار) داده ها درچند پردازشگرهای/ چند بخش ها برای کاربران آشکارنیست. این امر transparency (ناپیدایی= درقبل توضیح داده شد) نام می گیرد. تکنولوژی پایگاه داده موازی/ توزیعی مفهوم انتقال داده ها را که اصل مرکزی وبنیادین مدیریت پایگاه داده است برای محیط هایی که داده ها برروی تعدادی ازدستگاه های متصل با یک شبکه، توزیع و تکرارمی شوند، توسعه می دهد. این امرتوسط چندین شکل از« ناپیدایی» فراهم می شود: شبکه (ونیزناپیدای توزیع، ناپیدایی تکراروناپیدایی چند پارگی) دسترسی « ناپیدا» به این معنی است که کاربریک تصویرمنطقی یگانه ای را ازپایگاه داده می بیند، اگرچه این ممکن است به صورت فیزیکی توزیع شده باشد وآنها را قادرمی سازد که با پایگاه داده توزیعی دسترسی داشته باشند چنانکه گویی نوع متمرکزی بوده است. درشکل ایده آل آن، ناپیدایی کامل مستلزم یک رابط زبان جستجو برای

DBMS توزیع/ موازی می باشد که هیچ تفاوتی با آن DBMS متمرکزنداشته باشد.

مسائل ناپیدایی درمورد DBMS های توزیعی، بیشتربیان می شوند. دودلیل اساسی برای این امروجود دارد. اول ازهمه، سیستم چند پردازشگر که بریک DBMS اجرا می شود، توسط یک سیستم عامل منفرد ومنحصربه فرد کنترل می شود.

بنابراین سیستم عامل ،می تواند برای اجرای جنبه هایی ازعاملیت DBMS ساخته شود، ازاینرو درجایی ازناپیدایی را فراهم می آورد. دوما، توسعه نرم افزاردرسیستم های موازی توسط زبان های برنامه نویسی موازی پشتیبانی می شود که می توانند ناپیدایی (صنعتی برای یک وسیله، تابع یا بخشی ازبرنامه است که آنقدربه ملایمت ونرمی کارمی کند که برای کاربرقابل رویت نمی باشد. مثل توانایی یک برنامه دربه کارگیری فایل های ایجاد شده توسط یک برنامه دیگردرصورتی که کاربردرباز‌کردن، خواندن یا استفاده ازفایلهای برنامه دوم با شکلی مواجه نشده ویا اینکه اصلا متوجه آن شود، بیشتری را فراهم آورند. دریک DBMS توزیعی، داده ها وبرنامه های کاربردی که به آن داده ها دستیابی دارند، می توانند دربخشی مشابه مستقرشده، نیازبه دسترسی ازراه دوربه داده ها را که نوعی ازسیستم های اشتراک زمانی پردازش ازراه دور- مبنا می باشد، برطرف کند (یا کاهش دهد)، علاوه براین، ازآنجا که هربخش (سایت) برنامه های کاربردی کمتروکوچکتری ازپایگاه داده را اداره می کند، اتصال برای منابع وبرای دسترسی داده ها می تواند کاهش یابد. درنهایت موازی

گری ذاتی سیستم های توزیعی، امکان موازی گری درون جستجویی ودرجستجو را میسرمی سازد.

اگردسترسی کاربر به پایگاه داده توزیعی، تنها شامل جستجویی (پرس وجویی) (یعنی دسترسی فقط خواندنی) باشد، پس امکان موازی گرد، مستلزم آن است که بیشترپایگاه داده تا حد ممکن، تکراری باشد. هرچند، ازآنجا که بیشتردسترسی های پایگاه داده، فقط- خواندنی، نیستند، ترکیب عملیات های خواندن وبه روزرسانی برای تراکنش های توزیعی، به پشتیبانی نیازدارند. (که دربخش بعدی بررسی خواهند شد). کارایی بالاتر، احتمالا مهمترین هدف DBMS های موازی می باشد. دراین سیستم ها، کارایی بالاترمی تواند با چندین راه حل مکمل، حاصل گردد. پشتیبانی با سیستم عامل با جهت گیری پایگاه داده، موازی گری، بهینه سازی وتعادل بارگذاری، مقید کردن ودرفشارقراردادن سیستم عامل به و« آگاهی» ازملزومات پایگاه داده های خاص، (به عنوان مثال مدیریت بافر) اجرای کارکردهای سطح- پایین پایگاه داده را ساده کرده وبنابراین هزینه آنها را کاهش می دهد. برای مثال، هزینه یک پیام، می تواند به میزان قابل توجهی تا چند صد تعرفه قرارداد ارتباطات، کاهش یابد. موازی گری می تواند توان عملیاتی (استفاده ازموازی گری درجستجویی) را افزایش داده وزمان پاسخ تراکنش (استفاده ازموازی گری درجستجویی ودرون – جستجویی) را کاهش دهد.

DBMS های توزیعی وموازی برآن هستند که قابلیت اطمینان را بهبود بخشند زیرا آنها مولفه های تکراری (روی هم افتاده) دارند وبنابراین نقاط یگانه خرابی را حذف می کنند. خرابی یک بخش یا پردازشگرمنفرد، یا خطای یک اتصال ارتباطاتی که یک یا چند ساعت را غیرقابل دسترس می سازد، برای بازکردن کل سیستم، به صرفه نیست. این به آن معناست که اگرچه برخی داده ها ممکن است غیرقابل دسترسی باشند، پشتیبانی کردن ازتراکنش به اجرای کنترل توزیعی همزمان وقراردادهای پایایی (قابلیت اطمینان) توزیعی (به کارگیری وبازیابی) که دربخش بعد بررسی خواهند شد، نیازدارند.

دریک محیط توزیعی یا موازی، منطبق کردن افزایش اندازه پایگاه داده با افزایش تقاضاهای کارایی، می تواند آسانترباشد. پیاده سازی کامل قطعات سیستم های بزرگ، به ندرت ضرورت دارد، توسعه می تواند معمولا با اضافه کردن توان پردازش وذخیره سیستم، انجام گیرد. به صورت ایده آل، یک DBMS موازی (وتا درجه ای کمتر، یک DBMS توزیع) باید دومزیت را نشان دهد، افزایش مقیاس خطی وتسریع خطی. افزایش مقیاس خطی به یک کارایی ثابت (مداوم) برای افزایش خطی درهردومورد اندازه پایگاه داده وتوان ذخیره پردازش، بازمی گردد. تسریع خطی به یک افزایش خطی درکارایی برای یک اندازه پایگاه داده ثابت ویک افزایش خطی درتوان ذخیره وپردازش، بازمی گردد. علاوه براین توسعه سیستم باید به کمترین سازماندهی مجدد پایگاه داده موجود نیازداشته باشد. مشخصه های قیمت/ کارایی

میکروپروسسورها (ریزپردازنده ها) وایستگاه های کار، قراردادن یک سیستم ازکامپیوترهای کوچک با توان معادل یک دستگاه بزرگ منفرد (یگانه –تکی) را اقتصادی ترساخته اند. بسیاری ازDBMS های توزیعی تجاری درمینی، کامپیوترها وایستگاه های کار، به جهت سود بردن ازویژگی های قیمت/ کارایی مطلوب آنها، عمل می کنند. اتکای جاری به تکنولوژی ایستگاه کاربه انجام رسیده وروی داده است زیرا بیشترDBMS های توزیعی تجاری درون شبکه های حوزه محلی کارمی کنند که تکنولوژی ایستگاه کارآن ازهمه مناسب ترباشد.

ظهور DBMS های توزیعی که به شبکه های گسترده (یک شبکه ارتباطاتی که نواحی جداگانه جغرافیایی ودوری را به هم متصل می سازد) اجرا می شوند ممکن است اهمیت پردازنده مرکزی (سیستم اصلی- کامپیوتر بزرگ) را افزایش دهند. ازسوی دیگرDBMS های توزیعی آینده، ممکن است ازسازمانهای سلسله رانی پشتیبانی کنند که سایت های شامل گروههایی ازکامپیوترهای دریک شبکه حوزه محلی با یک شبکه گسترده بزرگ وسریع که گروه ها را بهم متصل می کند، باهم مرتبط اند.

تکنولوژی پایگاه داده موازی وتوزیعی:

DBMS های توزیعی وموازی، عاملیت مشابه DBMS های متمرکزدارند به جزمحیطی که داده ها درسایتهای به یک شبکه کامپیوتریا درگروه های یک سیستم چند پردازشگر، توزیع می شوند. همانگونه که دربالا بحث شد، کاربران ازتوزیع داده

ها، آگاه نیستند. بنابراین این سیستم ها نمایشی منطقا یکپارچه ازپایگـاه داده فیزیکـی توزیع شده (به صورت فیزیکی توزیع شده) عرضه مـی کننـد. برقـراری وحفـظ ایـن نمایش (نمایش درصفحه مانیتورکاربر) چالش های قابل ملاحظـه ای درکارکردهـای سیستم ایجاد می کند. ما بررسی اجمالی ازاین چالش های جدید دراین بخش ارائه می دهیم. ما فرض را براین می گیریم که خواننده با تکنیک های اصـولی واولیـه مـدیریت پایگاه داده آشنایی دارد.

موضوعات معماری

گزینه های توزیع ممکـن بسـیاری وجـود دارد. معمـاری (طراحـی) رایـج سـرویس گیرنده/ سرویس دهنده که تعدادی ازدستگاه های سرویس گیرنده به یک سـرورتکی پایگاه داده دسترسی می یابد، ساده ترین نوع آنهاسـت. دراین سیسـتم هـا کـه مـی توانند چند سرویس گیرنده / چند سـرویس دهنـده (سـرور) نامیـده شـوند، مسـائل مدیریت پایگاه داده به صورت قابل ملاحظه ای ساده می باشند زیرا پایگاه داده دریک تک سرور (سرویس دهنده) ذخیره می شود. موضوعات تکراری، مربوط به مـدیریت بافرهای سرویس گیرنده و حافظه نهایی داده ها ودرصورت امکان قفل ها می باشند. مدیریت داده به صورت مرکزی درتک سرور انجام می شود. یـک معمـاری تـوزیعی ترو منعطف ترمعماری چند سرویس گیرنده/ چند سرویس دهنده است که پایگاه داده درچند سرویس دهنده توزیع می شود کـه بایـد بـا یکدیگردرپاسـخ بـه جسـتجوهای کاربرودراجرای تراکنش های مرتبط باشند هردستگاه سرویس گیرنده یک سـرویس

دهنده (سرور) خانگی که درخواست ها ونیازهای کاربررا اداره می کند. ارتباط سرورها مابین خودشان برای کاربرناپیداست. بیشترسیستم های مدیریت پایگاه داده یک نوع یا انواع دیگرازمعماری های سرویس گیرنده- سرویس دهنده را اجرا می کنند.

یک DBMS توزیع صحیح، مابین دستگاه های سرور وسرویس گیرنده، تمایزوتفکیک قائل نمی شود. به صورت ایده آل، هرسایت (بخش) می تواند عاملیت یک سرویس گیرنده ویک سرویس دهنده را انجام دهد. چنین معماری های همتا (همراه به همراه) نامیده می شوند که نیازبه پروتکل(قرارداد) های پیچیده ای برای مدیریت داده های توزیع شده درچند بخش ها (چند سایت ها) دارد. ترکیب وپیچیدگی نرم افزارمورد نیاز، ارائه محصولات DBMS توزیعی نظیربه نظیررا با کندی وتاخیرهمراه کرده است. دامنه معماری های سیستم موازی بین دوکران، معماری بدون اشتراکی ومعماری حافظه تسهیم شده (حافظه اشتراکی)، میباشند. یک نقطه مفید مابین این دومعماری، معماری دیسک مشترک می باشد. درشیوه بدون اشتراک، هرپردازشگر، دسترسی انحصاری به حافظه اصلی وواحد ها دیسک دارد، ازاین رو هرگره می تواند به عنوان یک بخش محلی (درونی) با پایگاه داده ونرم افزار خودی، دریک سیستم پایگاه داده توزیعی مشاهده گردد. تفاوت میان DBMS های موازی بدون – اشتراک وDBMS های توزیعی، اصولا دریک پلات فرم (طرح زیربنایی- تکنولوژی اصلی سیستم کامپیوتری) اجرا است، بنابراین بیشترین راه حل

های طراحی شده برای پایگاه داده توزیعی می توانند درDBMS های موازی، بازیابی شوند. علاوه براین، معماری بدون – اشتراکی سه برتری عمده دارد: هزینه، توسعه پذیری ودسترسی، ازسوی دیگر، این معماری ازمتحمل مسائلی وچون پیچیدگی بیشتروتعادل بارگذاری (پتانسیل) می باشد.

نمونه هایی ازسیستم های پایگاه داده موازی بدون اشتراکی (بی اشتراکی) شامل محصولات DBS ترادف(teradetu) و spotsal , N.A tandenm ونیزتعدادی ازنسخه های اصلی (مدل نمونه) همچون BUBBA، GAMMA، PRISMA، ARBRE می باشند.

درشیوه حافظه تسهیم شده (حافظه اشتراکی) هرپردازشگر به هرمدول حافظه (واحد حافظه) یا واحد دیسک حافظه توسط یک رابطه سریع، دسترسی دارد (مثلا یک مسیرپرسرعت یا گزینه شطرنجی). چندین طراحی پردازنده مرکزی (سیستم اصلی) جدید مانند IBM ۳۰۹۰ یا Bull's DPSB وچند پردازشگرهای متعارف مانند sequent و encore ازاین شیوه تبعیت می کنند. معماری حافظه – اشتراک دومزیت عمده دارد: سادگی و تعادل بار. درمقابل این دومزیت، این مسائل نیزوجود دارد، هزینه، توسعه پذیری محدود ودسترسی پائین. نمونه هایی ازسیستم پایگاه داده موازی حافظه اشتراک شامل XPRS، DBS3 و Volcano ونیزدرگاه (صادرکردن) RDBMS های بزرگ برچند پردازشگرهای حافظه- اشتراکی. ازیک جهت، اجرای DB2، روی یک IBM۳۰۹۰ با ٦ پردازشگر، اولین نمونه بود، تمام محصولات تجاری

حافظه- تسهیم شده (اشتراکی) درامروزتنها ازموازی گری جستجوی استفاده می کنند.

درررویکرد دیسک- اشتراکی، هرپردازشگربه هرواحد دیسک به واسطه اتصال دسترسی دارد اما قسمت انحصاری (غیراشتراکی) به حافظه اصلی آن دسترسی دارد. هرپردازشگر می تواند به صفحات پایگاه داده دردیسک اشتراکی دسترسی داشته باشد ودرحافظه نهانی خود (یک سیستم حافظه فرعی ویژه که داده های با کاربرد زیاد را درخود جای می دهد تا دستیابی به آنها سریعترصورت گیرد). آنها را کپی کند برای جلوگیری ازناسازگاری ها وقاطی شدن دسترسی ها به صفحات مشابه قفلها وپروتکل ها سراسری برای حفظ ارتباط مطالب ووابستگی حافظه نهانی مورد نیازاست. دیسک اشتراکی ازمزیت هایی چند برخورداراست. هزینه، قابل گسترش، تعادل بار، دسترسی وانتقال آسان ازسیستم تک پرداشگری، ازسوی دیگر، دارای معایبی بودن پیچیدگی بیشترومسائل عملکردی بالقوه ای می باشد.

نمونه هایی ازDBMS موازی دیسک اشتراکی، IBM IMSها/ محصول اشتراک داده ها VS و DEC'S VAX DBMS و محصولات RdD می باشند. دراجرای ORACLE روی طبقه DEC'S VAX وکامپیوترهای NCUBE نیزازرویکرد دیسک- اشتراکی استفاده می شود زیرا به کمترین توسعه وگسترش هسته اصلی Kerael : بخش اصلی سیستم عامل که حافظه، فایل ها وابزارهای جانبی را اداره می

کند، نیازدارد. البته همه این سیستمها فقط ازموازی گری درجستجوی استفاده نمی کنند. پردازش وبهینه سازی پرس وجو (روند استخراج داده ها).

پردازش جستجو فرایندی است که طی آن یک جستجوی اظهاری (بیانی) به دوعملیات دستکاری داده های سطح پایین ترجمه وتعبیرمی شود. SQL زبان پرس وجوی استانداردی است که درDBMS های رایج پشتیبانی می شود. بهینه سازی پرس وجو به فرایندی اطلاق می شود که استراتژی برای بهترین برای یک پرس وجو ازمیان مجموعه ای ازگزینه ها، یافته می شود.

درDBMS های متمرکز، فرایند شامل دومرحله است: تجزیه وپرس وجو وبهینه سازی پرس وجو تجزیه پرس وجو یک SQL را می گیرد وآن را به چیزی تعبیرمی کند که درجبررابطه ای نشان داده می شود. دراین فرایند، پرس وجو، تحلیل می شود (ازنظرمعنایی) به طوریکه پرس وجوهای ناصحیح تعین شده وتا حد امکان پس زده می شوند وپرس و جوهای صحیح ساده سازی میشوند. ساده سازی (مختصرسازی) شامل حذف خبرهای اضافی است که ممکن است به عنوان نتیجه ای ازاصلاح وتعدیل پرس وجو برای رسیدگی به نظرات، اجرای امنیت وکنترل یکپارچگی معنایی، نشان داده شده ووارد شوند. پرس و جوی مختصرشده (ساده سازی شده) پس به عنوان یک پرس وجوی جبری بازسازی می شود برای یک پرس وجوی SQL ، چندین پرس وجوی ممکن وجود دارد. برخی ازاین پرس و جوهای جبری بهترازبقیه هستند. کیفیت یک پرس وجوی جبری (جبررابطه ای) برحسب عملکرد مورد

انتظارتعریف می شود. روش معمول، بدست آوردن یک پرس وجوی اولیه با ترجمه وتعبیربیان ها (جملات) وهدف گذاری درعملیات رابطه ای است که درپرس وجوی خبری اولیه پس با استفاده ازقوانین تبدیل جبری به دیگرپرس وجوهای جبری تبدیل می شود تا بهترین آن یافته شود. بهترین پرس وجوی جبری طبق یک تابع هزینه که هزینه اجرای پرس و جو را مطابق آن تخصیص جبری محاسبه می کند، تعین میشود. این فرایند بهینه سازی پرس وجواست.

درDBMS های توزیعی، دومرحله بیشترمیان تجزیه پرس وجو وبهینه سازی جستجو وجوددارد: محلی سازی داده ها وبهینه سازی سراسری (جهانی) جستجو ورودی برای محلی سازی داده ها، پرس وجوی جبری اولیه ای است که مرحله تجزیه جستجو به وجود آمده است. پرس وجوی جبری اولیه به رابطه ای سراسری بدون توجه به فراگمت یا توزیع آنها، تخصیص داده می شود. نقش اصلی محلی سازی داده ها، محلی کردن داده های پرس و جو با استفاده ازاطلاعات توزیع داده است. دراین مرحله، فراگمنت ها که درپرس وجوقراردارند، تعیین شده وپرس وجو به آن چه که درفراگمنت ها بیش ازرابطه های سراسری عمل می کند، تبدیل می شود. هماهنگونه که درقبل نشان داده شد، بشخ بخش سازی توسط قوانی بخش بخش بازی تعریف می شود که می توانند به عنوان عملیات رابطه ای نشان داده شوند (پراکندگی افقی طبق انتخاب، پراکندگی (بخش بخش کردن) طبق طرح). یک رابطه توزیعی می تواند با به کارگیری برخی ازقوانین بخش بخش سازی، بازسازی شود.

این کاربرنامه محلی سازی نامیده می شود. برنامه محلی سازی بریک پرس وجوی بخش بخش شده افقی (عمودی)، اجتماع (اشتراک) فراگمنت ها (بخش ها) است. ازاین رو درفلان مرحله محلی سازی داده ها، هررابطه سراسری (عمومی) درابتدا با برنامه محلی آن جایگزین می شود وسپس پرس وجوی فراگمنت بدست آمده، ساده شده وبرای ایجاد پرس وجوی خوب دیگری بازسازی می شود. ساده سازی (مختصرسازی) وبازسازی می تواند مطابق قوانین مشابه به کارگرفته شده درمرحله تجزیه (بازکردن مطالب) انجام پذیرند.

درمرحله تجزیه، پرس وجوی فراگمنت نهایی، عموما، هنوزحالت بهینه فاصله دارد تا دراین فرایند فقط پرس وجوهای جبری « بد» حذف می شوند.

ورودی مرحله سوم یک پرس وجوی فراگمنت شده (بخش بخش شده) می باشد که به صورتی است که یک پرس و جوی جبری روی فراگمنت ها وجود دارد. هدف ازبهینه سازی پرس وجو، یافتن یک استراتژی اجرایی برای پرس وجواست که به حالت بهینه نزدیک است، به یاد داشته باشید که یافتن راه حل مطلوب وبهینه ازنظرمحاسباتی کاری سخت وغیرممکن است. یک استراتژی اجرا برای یک پرس وجوی توزیعی می تواند با عملیات های جبر رابطه ای (درمدیریت پایگاه داده ها، مجموعه ای ازقواعد وعملگرهایی است که تغیرودستکاری رابطه ها (جداول) را ممکن می سازند جبررابطه ای معمولا دارای این عملگرها میباشد SELECT، PRODECT، PRODUCT، UNION وبا استفاده ازجبررابطه ای می توان رابطه

ها یا جداول جدیدی ازروی رابطه های موجود درپایگاه داده ها ساخت. وعناصراولیه ارتباطی (عملیات ارسال/ دریافت) برای انتقال داده ها میان سایت ها، توصیف شـود. هرچند این بهینه سازی، مستقل ازمشخصه های فراگمنت، همچون اعـداد اصـلی مـی باشد. به علاوه، عملیات های ارتباط هنوزتخصیص داده نشده اند. با پس وپیش کردن وتبدیل ترتیب عملیات ها دریک پرس وجوی فراگمنت، بسیاری ازطـرح هـای اجرایـی پرس وجوی معادل، ممکن است یافت شوند. بهینه سازی پرس وجو که شـامل یـافتن مهمترین گزینه ازمیان طرح های نامزد این امـر، توسـط یـک بهینـه سـازآزموده مـی شود. یک بهینه کننده پرس وجو معمـولا بـه سـه مولفـه توجـه مـی کنـد، یـک فضـی جستجو، یک مدل هزینه ویک استراتژی جستجو، فضای جستجو، مجموعه ای ازطرح های اجرایی برای پرس وجوی ورودی یم باشد. این طرح ها با توجه بـه اینکـه نتـایج مشابهی را حاصل می کنند، با هـم معـادل وبرابرنـد امـا درترتیـب اجرایـی عملیاتهـا وروشی که این عملیاتها طی آن انجام می شوند، متفاوتند. مدل هزینه، هزینه یک طرح اجرایی داده شده را پیش بینی می کند. به عبارت دقیق تر، مدل هزینه، باید دانش دقیق و کاملی درباره محـیط اجـرای مـوازی داشـته باشـد. اسـتراتژی جسـتجوی فضـای جستجو را بررسی کرده وبهترین طرح را برمی گزیند. این استراتژی تعریف می کنـد که کدام طرح درکدام ترتیب باید آزموده شود.

دریک محیط توزیعی، تابع هزینه که اغلب برحسب واحدهای زمانی تعریف می شـود، به منابع محاسباتی همچون فضای دیسـک، ؟ دیسـک، فضـای بـافر(حافظـه میـانی)،

هزینه CPU، هزینه ارتباط و... بازمی گردد. معمولا این تابع، ترکیبی وزنی ازهزینه های ؟ ، CPU وارتباط است. با این وجود، دریک ساده سازی شاخصی که توسط DBMS های توزیعی صورت می گیرد، هزینه ارتباط به عنوان مهمترین عامل مورد ملاحظه قرارمی گیرد. این برای شبکه های منطقه ای وسیعی نیزتایید می شود، چرا که پهنای باند محدود، ارتباط را بسیارگرانترازآن درپردازش محلی، می سازد. برای انتخاب وگزینش ترتیب عملیاتها، ضروری است که هزینه های اجرای مرتبط های نامزد شده (مرتبه های کاندید شده) پیش بینی شوند. تعیین هزینه های اجرا پیش ازاجرای پرس و جو (یعنی بهینه ساز استاتیک) برمبنای آمارفراگمنت وفرمولهای برآورد موارد اصلی نتایج عملیات های رابطه ای است. ازاین رو تصمیم های بهینه سازی به آمارفراگمنت های موجود بستگی دارند. جنبه مهمی ازبهینه سازی پرس وجو ترتیب پیوند است، زیرا پس وپیش کردن وجایگشت پیوندها (join= پیوند= یک عملیات جدول پایگاه داده ها که درنتیجه تطبیق فیلد کلیدی یک جدول با ورودی یک جدول دیگر، پیوندی بین آن دو برقرارمی شود.) می تواند منجربه بهبود چند مرتبه وترتیب بزرگی گردد. یک تکنیک اصولی برای بهینه سازی یک توالی عملیات های پیوند توزیع، استفاده ازعملگرنیمه پیوند است. کاراصلی نیمه پیوند دریک سیستم توزیعی، کاهش اندازه عملوندهای (مفعول یک عملیات ریاضی یا دستورالعمل های کامپیوتری که می توانند یک مورد داده ای یا مکان ذخیره داده ها (درحافظه یا دیسک) باشد به عنوان مثال درعمل ۳+۲ ، عددهای ۲ و ۳ عملوندهای عملیات جمع

هستند. پیوند وبنابراین، کاهش هزینه ارتباط است. هرچند تکنیک های جدیدترکه پردازش محلی هزینه های ارتباط را بررسی می کنند ازنیمه پیوندها استفاده نمی کنند زیرا ممکن است جبری بهینه شده است که عملیات های ارتباط درفراگمنت های قرار گرفته اند. بهینه سازی پرس وجوی موازی، شبیه به پردازش پرس وجوی توزیع عمل می کند.

این ازمزیت موازی گری درون عملیاتی، که قبلا بیان گردید ونیزموازی گری درعملیات برخورداراست. موازی گری درون عملیاتی با اجرای یک عملیات درچندین گره یک دستگاه چند پردازشگربه دست می آید. این امربه آن نیازدارد که عملوندها ازقبل پارتیشن بندی (تفکیک وجزء جزء) شده باشند به عنوان مثال طی آن افرازمی شود (تقسیم بندی) موضوع یک طراحی فیزیکی (این اصطلاح به مفهوم جنبه واقعی است وبه نحوی نیزبه سخت افزاربط دارد. درمقابل این اصطلاح، اصطلاح منطقی قراردارد. که وجود آن مفهوم بوده اما الزاما فیزیکی نیست) است. نوعا، افراز(پارتیشن بندی) با به کارگیری یک تابع وکاردرهم (آمیخته- ظهور کاراکترهای درهم وبرهم درصفحه نمایش) درباره یک ویژگی، صفت یک رابطه، که اغلب صفت مشترک می باشد، انجام می پذیرد. مجموعه گره هایی که یک رابطه آنجا ذخیره می شود، home (خانه) آن رابطه نامیده می شود. خانه یک عملیات، مجموعه ای ازگره هاست که عملیات آنجا اجرا می شود وباید خانه عملوندهای آن نیزباشد تا عملیات بتواند برعملوندهایش دسترسی داشته باشد. برای عملیات باینری (دودویی). همچون join

درقبل توضیح داده شد. این خانه باید برافراز مجدد یکی ازعملوندها دلالت داشته باشد. بهینه ساز حتی گاهی به این نتیجه می رسد که افرازمجدد هردوعملوند، مفید است. بهینه سازی موازی برای استفاده کردن وبهره برداری ازموازی گری درون عملیاتی می تواند برخی ازتکنیک ها برای استفاده پایگاه داده های توزیعی برجای بگذارد. موازی گری درعملیات زمانی روی می دهد که دو یا چند عملیات به صورت موازی همه به صورت جریان داده وهم به صورت مستقل، اجرا شوند. ما جریان داده را شکلی ازموازی گری القاء شده توسط اجرای ارتباطی روش استخراج وکد برگردانی دستورالعمل ها به نحوی که دریک مدت زمان خاص چندین دستورالعمل برنامه درمراحل مختلف، استخراج وبرگردانی شوند تعریف می کنیم. موازی گری مستقل زمانی روی می دهد که عملیات ها دریک زمان یا با ترتیبی اختیاری ولخواهی اجرا شوند. موازی گری مستقل تنها زمانی است که عملیات ها شامل داده های مشابه نباشند.

کنترل همزمانی

هرماه چندین کاربربه یک پایگاه داده مشترک دسترسی(خواندنی ونوشتنی) دارند، این دسترسی ها به هماهنگی به جهت تضمین وبرآورده شدن همسانی وهماهنگی پایگاه داده، نیازدارند. هماهنگی، (هماهنگی ومطابقت زمانی) به وسیله الگوریتم های کنترل همزمانی به دست می آید که یک معیاردرستی ای (صحت) همچون قابلیت تسهیل را اعمال می کنند. دسترسی کاربران برحسب تراکنش ها ازهم منفک می شود

که پائین ترین سطوح کاری این تراکنش ها مجموعه ای ازعملیات های خواندن ونوشتن برپایگاه داده می باشد. الگوریتم های کنترل همزمانی، ویژگی ایزوله سازی (جداسازی) را دراجرای تراکنش اعمال می کنند که به این معنی است که اثرات یک تراکنش برپایگاه داد ازدیگرتراکنش ها جداسازی می شود تا زمانیکه اجرای اولین تراکنش کامل شده وبه پایان برسد.

رایج ترین الگوریتم های کنترل همزمانی مبنایی قفل کننده دارند (قفل کننده= locking= روند جلوگیری ازاستفاده یک فایل یا رکورد، پایگاه داده ها بخصوص درشبکه ها ووضعیت های مشابه که ممکن است بیش ازیک نفربطورهمزمان اقدام به استفاده ازیک فایل مشترک یا تغییررکورد یک پایگاه داده ها بکنند) درچنین طرح هایی، یک قفل، چه درحالت اشتراکی چه درحالت انحصاری، بروی برخی واحدهای ذخیره (معمولا یک صفحه) قرارداده می شود (زمانیکه یک تراکنش قصد دسترسی به آن رادارد). این قفلها، طبق قواعد قفل خواندن- نوشتن، نوشتن- خواندن، نوشتن- نوشتن، قرارداده می شوند تا ازتعارض ها وبرخوردهای میانه جلوگیری به عمل آید. قاعده مشهوروشناخته شده ای است که اگرعمل قفل گذاری روی تراکنش های همزمان ازاین قانونه ساده پیروی کنند، تصمین تسلسل این تراکنش ها ممکن می گردد، تراکنشی نباید بروی آن قفل گذاشته شود که درقبل یکباربروی آن برتراکنش آزاد دیگر، ازآن جلوگیری شده است (قفل شده است). این قانون به قفل گذاری دوفازی (مرحله ای) مشهوراست زیرا تراکنش هایی که قف شده اند (ازاجرای آن

ممانعت شده است) وارد یک فاز(مرحله) افزایشی (بازشدن قفل) شده وهنگامی که قفل آنها بازمی شود وارد فازکاهشی (قفل شدن) می شوند.

ازاین رو بیشترالگوریتم های کنترل همزمانی قفل- مبنا، اکید ودقیق می باشند چرا که تا پایان یک تراکنش، تراکنش دیگررا قفل نگه می دارند.

درDBMS های توزیع، چالش پیش رو بسط وتوسعه به قابلیت تسلسل و آرگومان الگوریتم های کنترل همزمانی محیط اجرای توزیعی است. دراین سیستم ها، عملیات های یک تراکنش ممکن است درچندین سایت که به داده های دارنده، اجرا شوند. درچنین موردی، اعمال وتخصیص قابلیت تسلسل آرگومان، مشکل تراست. درآمیختگی، ناشی ازاین حقیقت است که ترتیب تسلسل مجموعه ای مشابه ازتراکنش ها ممکن است درسایت های مختلف ، متفاوت باشد، بنابراین، اجرای مجموعه ای ازتراکنش های توزیعی تسلسل پذیراست اگروتنها اگر:

۱- اجرای مجموعه ای ازتراکنش ها درهرسایت تسلسل پذیرباشد.

۲- ترتیب های تسلسل این تراکنش ها درتمام این سایت ها یکسان باشند.

الگوریتم های کنترل همزمانی توزیعی یک تسلسل سراسری را اعمال می کنند. درالگوریتم های قفل – مبنا سه شیوه گزینشی ازاجرای تسلسل سراسری (کلی) وجود دارد: قفل گذاری متمرکز، کپی اصلی والگوریتم قفل گذاری توزیعی. درقفل گذاری تمرکز، یک جدول قفل گذاری منحصربه فرد برای سراسرپایگاه داده توزیعی وجود دارد. این جدول قفل گذاری، دریکی ازسایتها (بخش ها) تحت کنترل یک

مدیرقفل گذاری قرارمی گیـرد. مـدیرقفل گـذاری، مسـئول قـراردادن وبـازکردن قفـل برتراکنش هاست. ازآنجاکه تمام قفل ها دریک سایت (بخش) مدیریت می شوند، ایـن شیوه مشابه کنتـرل متمرکزهمزمـانی اسـت وبـرای اعمـال واجـرای قـانون مسلسـل سراسری ما آسان است. اجرای این الگوریتم هـا آسـان اسـت امـا ایـن الگـوریتم هـا دومسئله، درمضیقه اند. سایت مرکـزی ممکـن اسـت بـه دلیـل تعـداد کارانجـام شـده وترافیکی که پیرامون آن ایجاد می شـود، درمضـیقه افتـاده ومحـدود کننـده عملیـات باشد، وسیستم ممکن است پایایی واعتبارکمتری داشته باشد زیرا نقـص ودسترسـی ناپذیری سایت مرکزی، باعث عدم دستریس سیسـتم خواهـد شـد. قفـل گـذاری کپـی اصلی، الگوریتم کنترل همزمانی است که درپایگـاه داده هـای تکرارشـونده ای (روی هم افتاده ای) که ممکن است چندین کپی ازیک مورد داده ای یک واحـد ازداده هـا کـه عنصرداده ای نیزنامیده می شود که درسایتهای مختلفـی ذخیـره مـی شـوند، وجـود دارد.

یکی ازکپی ها به عنوان کپی اصلی معرفی شده واین کپی است که ازدسترسی بـه آن باید ممانعت به عمل آید(قفل شود). مجموعه ای ازکپی های اصلی برای هرمورد داده برای تمامی سایت ها درسیستم توزیعی شناخته شده است وقفل تراکنش ها با جهـت گیری به کپـی اصـلی مناسـب آن انجـام میشـود. اگرپایگـاه داده هـای تـوزیعی، بـی اثرمیشود. قفل گذاری کپی اصلی برای نسخه توزیعی اصلی INGRES، درنظرگرفته شده بود. درقفل گذاری توزیعی (یا غیرمتمرکز) وظیفه مـدیریت واداره قفـل، درتمـام

سایت درسیستم، به اشتراک گذاشته می شود. اجرای تراکنش، مشـارکت وهمـاهنگی اداره کنندگان قفل را دربیش ازیک ساعت دربردارد. قفلهایی درهرسایت حاصـل مـی شوند که درآنجا تراکنش به یک مـورد داده (عنصـرداده) دسترسـی دارد. الگـوریتم های قفل گذاری توزیعی، هزینه اضافی نوع قفل گذاری متمرکزرا ندارند. هرچند هزینه ارتباطی برای گذاردن تمامی قفلها وپیچیدگی الگوریتم، بیشتراست. الگوریتم های قفـل گذاری وتوزیع درسیستم R ودرSQL یک سری به کارمی روند. یک اثرجانبی تمامی الگوریتم های کنترل همزمانی قفل- مبنا آن است که همگی آنها باعث ایجاد این ثبـت (وقفه = deadlock= وضعیتی که درآن دوبرنامه یا ابزارمنتظرند تا قبل ازادامـه کـار، پاسخی ازدیگری دریافت کنند) می شوند. تشخیص ومدیریت وقفه ها (بن بسـت هـا) دریک سیستم توزیعی، دشواراست. با این وجود سادگی نسبی وعملکرد بهترالگوریتم های قفل آنها را مهبوب ترورایج ترازگزینه های چون الگوریتم های نشان زمان- مبنا یا کنترل همزمانی خوش بینانه ساخته است. الگوریتم های نشان زمان مبنا با عملیات های متعارض ودرهم تراکنش ها، مطابق نشان (مهرهای) زمانی آنها عمل می کنند به گونه ای که این نشان ها پس ازاینکه تراکنش های پذیرفتـه شـدند، تعـین وتخصـیص داده می شوند. الگوریتم کنترل همزمانی خوش بینانه، با این مقدمه کـارمی کننـد کـه تداخل وتعارض تراکنش ها به ندرت روی می دهد و دراجرای تراکنش ها، به گونه ای عمل می کنند که تراکنش تا پایان صورت پذیرد، زمانیکـه درنقطـه ای خـاص تصـادم وتعارض تایید شود. اگراین تایید نشـان دهـد کـه قابلیـت تسلسـل بـا اتمـام موفقیـت

آمیزآن تراکنش خاص به خطرمی افتد، پس این تراکنش ناقص مانده وری استارت (شروع مجدد) میشود.

پروتکل های پایانی (اعتبار)

بیشترنشان دادیم که DBMS های توزیعی به صورت بالقوه ای پایاترند زیرا چندین مولفه سیستم دارند که نقص وخرابی یگانه را برطرف می کنند. این امرمستلزم سیستم طراحی دقیق واجرای واعمال پروتکل هایی برای رفع نقص سیستم است. دریک DBMS توزیعی، امکان چهارنوع خرابی (نقص) را داراست: خرابی تراکنش، خرابی های سایت (سیستم)، خرابی های رسانه (دیسک)، وخرابی های خط ارتباط. تراکنش ها به چندین دلیل می توانند بد عمل کرده وخراب شوند. خرابی میتواند ناشی ازیک خطا درتراکنش باشد که توسط داده ورودی ایجاد شده است ونیزمی تواند ناشی ازخطا درتعین یک بن بست (وقفه) بالقوه یا موجود باشد رویکرد معمول برای رفع این موارد، جلوگیری ازاتمام واکنش وراه اندازی مجدد (ری ا؟) به حالت پیش ازشروع پایگاه داده می باشد.

خرابی سایت (یاسیستم) ناشی ازخرابی (نقص) سخت افزار یعنی پردازشگر، حافظه اصلی، منبع تغذیه یا نرم افزار(اشکال درسیستم یا کد برنامه) می باشد. اثرخرابی سیستم، ازدست دادن محتویات حافظه اصلی است. بنابراین هربروزرسانی (نوسازی) قسمت های پایگاه داده که دربافرهای حافظه اصلی پایگاه داده فرارنیزنامیده می شود، هستند، درنتیجه خرابی سیستم، ازدست می روند. هرچند

پایگاه داده ای که درمنبع ذخیره ثانویه (پایگاه داده پایدارنیزنامیده می شود) ذخیره شده است صحیح وایمن باقی می ماند. برای دستیابی به این امر، DBMS ها نوعا، پروتکل های ثبت وقایع Logging protocol همچون write-Aheadlogging را به کارمی گیرند که تغیرات پایگاه داده را درلوگ های سیستم (شرح ماوقع سیستم- گزارش روزانه عملیات سیستم) ثبت کرده واین ثبت ها وصفحات پایگاه داده فرارا به حافظه ثابت (منبع ذخیره پایدار) درزمان مناسب انتقال می دهد. ازدور، او چشم اندازاجرای تراکنش توزیع، خرابی های سایت مهم می باشند زیرا سایت های خراب (دارای نقص) نمی توانند دراجرای هیچ تراکنشی شرکت کنند.

خرابی رسانه به خرابی ابزار منبع ذخیره ثانویه که پایگاه داده پایدار(ثابت) را ذخیره می کند، باز می گردد. نوعا، این خرابی ها ابزارهای منبع ذخیره دوپلکس وحفظ کپی آرشیوی ازپایگاه داده نسبت داده می شوند. خرابی های رسانه مختص به مسائلی داخلی یک سایت بوده وبنابراین چه مکانیسم پایایی (اعتبار) و DBM های توزیعی نسبت داده نمی شوند. سه نوع خرابی توصیف شده دربالا برای هردو DBMS توزیع ومتمرکز، متداول اند. خرابی های ارتباط، ازسوی دیگر، مختص به سیستم های توزیعی است. چند نوع خرابی ارتباط وجود دارد. رایج ترین آنها خطا درپیغام، ترتیب نادرست پیام ها ازدست رفتن (یا تحویل داده نشدن) پیام ها وخرابی های خطا می باشند. معمولا مسئولیت دوتای اول با پروتکل های DBMSتوزیعی دارند وبنابراین نیازاست که درطراحی این پروتکل ها ملاحظات صورت گیرد. اگریک سایت

(بخش) منتظرپیغامی ازسایت دیگراست واین پیغام هرگزنرسد، این امربه دلیل آن است که:

a- پیغام ازدست رفته است.

b- خطی (خطوطی) که دوسایت را به هم متصل می کند قطع شده است.

c- سایتی که تصورمی شد پیغام را ارسال کرده ممکن است دچارنقص یا خرابی شده است بنابراین همیشه ممکن نیست که میان خرابی های سایت وخرابی های ارتباط تمیزقائل شویم. سایت درحال انتظاربه سادگی وقفه دارد وباید اینگونه پنداشت که سایت دیگربدون ارتباط است.

پروتکل های DBMS توزیعی باید برای چنین عدم قطعیتی، وقف یابند. یک نتیجه گیری موثرازخرابی های خط می تواند افرازشبکه (پارتیشن بندی) (تکنیک) سایتها ازگروه ها باشد که ارتباط درون هرگروه ممکن است اما درارتباط بین پروتکل های DBMS توزیعی باید برای چنین عدم قطعیتی، وقف یابند. یک نتیجه گیری موثرازخرابی های خط می تواند افرازشبکه (پارتیشن بندی) (تفکیک) سایتها ازگروه ها باشد که ارتباط درون هرگروه ممکن است اما ارتباط بین گروه ها ممکن نیست. پرداختن به این مفهوم که دردسترس ساختن پایگاه داده برای دستیابی به آن ودرهمان زمان تضمین سازگاری وهمسانی آن می تواند ممکن نباشد، مشکل است. دوویژگی تراکنش ها با پروتکل های موثق حفظ می شوند: تجزیه ناپذیری ودوام.

تجزیه ناپذیری مستلزم آن است که با همه عملیات های یک تراکنش اجرا شوند یا هیچ یک اجرا نگردند.

همه یا هیچ کدام، بنابراین مجموعه عملیات های موجود دریک تراکنش به عنوان یک واحد یکپارچه (تجزیه ناپذیر) اجرا می شوند. تجزیه ناپذیری حتی درهنگام خرابی، حفظ می شود. دوام مستلزم آن است که تاثیرات اجرای کامل وموفق تراکنش، خاطاها وخرابی های بعدی را تاب آوردند.

اعمال واجرای دوام وتجزیه ناپذیری مستلزم اجرای پروتکل های الزامی تجزیه ناپذیری وپروتکل های بازیابی توزیعی می باشد. رایج ترین پروتکل اجرایی، انجام دوفازی (مرحله) می باشد. پروتکل های قابل بازیابی دربالاترین قله پروتکل های بازیابی محلی قرارمی گیرند که الزام تجزیه ناپذیری تراکنش های توزیعی را تضمین می کنند. انجام دومرحله ای (tpc) ساده ترین وعالی ترین مترجم برای برنامه ها یا الگوریتم های ارزشمند که ازسادگی، وکارایی ودقت مطلوبی برخوردارهستند که الزام تجزیه ناپذیری محلی تا تراکنشهای توزیعی گسترده است که برآن تاکید می ورزد که تمام سایتهایی که دارای اجرای یک تراکنش توزیعی هستند وبرانجام یک تراکنش پیش ازاینکه عوامل ونتایج آن پایدارشوند، ملزم باشند.

اگرتمامی سایتها (بخش ها) برانجام یک تراکنش پس ازآنکه تمامی کارهای تراکنش توزیعی اثربخش بود وکارگرافتار، توافق کنند، اگریکی ازسایتها برای انجام عملیات ها

درآن سایت با کاهش ونقصان همراه بود، تمامی سایتهای دیگرمستلزم لغو کردن تراکنش هستند بنابراین قانون tpc بیان می دارد که

۱- اگرحتی یک سایت انجام تراکنش را نپذیرفت (که به معنای آن است که آن سایت برلغو تراکنش رای دهد) تراکنش توزیعی باید درهرسایت که آن را اجرا می کند لغو گردد، و

۲- اگرتمامی سایتها برانجام تراکنش رای دهند، تراکنش توزیعی درهرسایتی که آن را اجرا می کند، انجام می شود. اجرای ساده پروتکل tpc مطابق زیراست. یک فرایند هماهنگ کننده درسایت که تراکنش های توزیعی ازآنجا سرچشمه می گیرند و فرایندهای مشترک درتمامی سایت های دیگری که تراکنش آنجا اجرا می شود، وجود دارند. درابتدا هماهنگ کننده یک پیغام آمادگی به تمامی شرکت کنندگان که هریک به صورت مستقل تعین می کنند که آیا تراکنش می تواند درسایت انجام شود، می فرستد. آنهایی که می توانند اجرا کنند، یک پیغام رای- اجرا پاسخ می دهند وآنها که قادربه اجرا نیستند یک پیغام رای- لغو ارسال می دارند.

هنگامی که شرکت کننده رای خود را رجیستر(ثبت) می کند، دیگرنمی تواند آن را تغیردهد. هماهنگ کننده، پیام ها را گرد آوری می کند وبه انجام یا عدم انجام تراکنش مطابق قانونtpc تصمیم می گیرد. اگرتصمیم برابر۱ باشد، هماهنگ کننده یک پیغام اجرای – سراسری به تمام شرکت کنندگان می فرستد، اگرتصمیم برلغو باشد، یک

پیغام ده اجرای- سراسری به تمام شرکت کنندگان می فرستد، اگرتصمیم برلغو باشد یک پیغام لغو-سراسری ارسال میدارد نیازبه هیچ پیغامی برای ارسال به آن شرکت کنندگانی که دراصل وآغازبه لغو رای داده اند، نیست زیرا آنها می توانند متصورشدند، مطابق قانون tpc که تراکنش سرانجام به طورسراسری لغو خواهد شد. این امرگزینه لغو یک طرفه شرکت کنندگان نامیده می شود. دوگروه ازپیغامها میان هماهنگ کننده وشرکت کنندگان مبادله می شود، ازاین رواین پروتکل به نام پروتکل tpc نامیده می شود. چند نوع tpc وجود دارد مانند tpc خطی و tpc توزیعی که فروشندگان DBMS توزیعی علاقه چندانی به آنها نشان نداده اند. دوگونه مهم tpc، با فرض (مسلم کردن) لغو و tpc با فرض انجام میباشند. این گونه ها ازاهمیت زیادی برخوردارند. زیرا آنها پیغام هزینه ؟ پروتکل را کاهش می دهند. پروتکل قطع مسلم شده دراستاندارد X/OPEN/ XA دخیل بوده وبه عنوان نیمی ازاستاندارد ISO برای پردازش توزیعی بار، سازگارشده است. یکی ازمشخصه های مهم پروتکل tpc ماهیت بلاک کردن (وقفه – ایست) آن است. خرابی ها درخلال فرایند اجرا، می توانند به وجود آیند. همانگونه که دربالاذکرشد، تنها را برای تشخیص این خرابیها وقفه ای است که درفرایند انتظاربرای پیغام، روی می دهد. زمانیکه این وقفه وفاصله روی می دهد، فرایند (هماهنگ کننده با شرکت کننده) آن وقفه ازیک پروتکل انقضاء برای تشخیص اینکه چه امری باید برای تراکنشی که درنیمه فرایند انجام است، صورت پذیرد، پیروی می کند. یک پروتکل اجرای بدون وقفه آن است که طی آن،

پروتکل انقضاء می تواند تعیین کند که درمورد خرابیها، چه تصمیمی برای تراکنش بگیرد. درمورد tpc، اگرخرابی سایت، درسایت هماهنگ کننده ویک سایت شرکت کننده درحالیکه هماهنگ کننده درحال جمع آوری آراء شرکت کنندگان را تشخیص دهند وآنها باید تا زمانیکه هماهنگ کننده یا شرکت کننده ای که قطع شده، بازیابی میشوند، بلوکه باقی بمانند. درخلال این پریود، قفلهای که برتراکنش قرارداده شده اند نمی توانند بازشوند، که این امردسترس به پایگاه داده ها را کاهش می دهد. تصویرکنید که یک شرکت کننده پس ازآنکه رای خود را به هماهنگ کننده ارسال کرد، پیش ازدریافت تصمیم نهایی هماهنگ کننده وقفه (قطع ارتباط) ایجاد گردد. دراین مورد گفته می شود که شرکت کننده درحالت READY(آماده) قراردارد. پروتکل انقضاء برای شرکت کننده مطابق زیراست. درابتدا متذکرمی شویم که شرکت کننده نمی تواند به صورت دلخواهی واختیاری تصمیم برلغو وانقضای تراکنشی بگیرد. ازآنجا که او درحالت آماده قراردارد، باید به اجرای تراکنش رای دهد. بنابراین اکنون نمی تواند رای خود را عوض کند وآن را لغو کند. ازسوی دیگر، او نمی تواند به صورت دلخواهی تصمیم به اجرای تراکنش بگیرد زیرا ممکن است که شرکت کننده دیگری به لغو آن رای داده باشد.

دراین مورد، شرکت کننده درحالت بلوکه (وقفه، قطع) خواهد ماند تا اینکه ازکسی (یا هماهنگ کننده یا دیگرشرکت کنندگان) سرانجام نهایی ووضعیت قطعی تراکنش را بفهمد. اگرمایک ساختارارتباط متمرکزرا که طی آن شرکت کنندگان نمی توانند با

یکدیگرارتباط داشته باشند مدنظرقراردهیم، شرکت کننده ای که درحالت قطع (وقفه- تایم اوت) قراردارد، باید منتظربماند تا هماهنگ کننده تصمیم نهایی درمورد تراکنش را به او گزارش کند. زمانیکه هماهنگ کننده دارای خرابی ونقص ارتباطی است، شرکت کننده درحالت بلوکه باقی خواهد ماند. دراین مورد هیچ پروتکل منطقی وتوجیه پذیری نمی تواند طرح شود.

اگرشرکت کنندگان بتوانند با یکدیگرارتباط داشته باشند، یک پروتکل انقضاء (لغو) توزیعی می توانند توسعه یابد. شرکت کنندگان که درتایم اوت قراردارند می توانند به سادگی ازدیگران شرکت کنندگانه برای اطلاع ازتصمیم اخذ شده، سوال کند، اگردرخلال انقضاء، تمامی شرکت کنندگان بفهمند که فقط سایت هماهنگ کننده خرابی (نقص) دارد، آنها می توانند یک هماهنگ کننده جدید انتخاب کنند که بتواند فرایند اجرا را ازنوراه اندازی (ری استارت) کند. هرچند درموردی که سایت هماهنگ کننده وسایت شرکت کننده خرابی دارد، برای شرکت کننده ای که خرابی سایت دارد، این امکان وجود دارد که تصمیم هماهنگ کننده رادریافت کرده باشد و طبق آن تراکنش را لغو کرده باشد. دیگرشرکت کنندگان ازاین تصمیم آگاهی ندارند، بنابراین اگرآنها یک هماهنگ کننده جدید انتخاب کنند و کاررا پیش ببرند، این خطروجود دارد که تصمیم برانقضای تراکنش بگیرند که این تصمیم با شرکت کننده ای که با خرابی سایت وقطع سایت مواجه است مغایرت داشته باشد. مورد بالا ماهیت blocking (بلوکه کردن- قطع جهان- انسداد) tpc را نشان می دهد. تلاشهایی برای ایجاد پروتکل

های اجرایی غیربلوکه (یعنی اجرای سه فـاز) صـورت گرفتـه امـا هزینـه بـالای ایـن پروتکل ها ازاین امرممانعت به عمل می آورد. معکوس وبرعکس انقضاء (پایـان دادن – قطع کردن) ، بازیابی است. هنگامی که سایت هـایی کـه قطـع بـوده وخـراب اسـت، بازیابی می شود، چه کاریایی باید برای بازیابی پایگاه داده درسایت برای یـک حالـت همسان وسازگان صورت پذیرد، این موضوع وحوزه پروتکل های بازیابی اسـت. بـا توجه به بخش ریکاوری (بازیابی) مورد بحث باشـد، ؟ کـه طـی آن سـایت هماهنـگ کننده ، عمل بازیابی را انجام می دهد پروتکل های بازیابی، اکنون باید تعیین کنند کـه درمورد تراکنش های توزیعی برای هماهنگی دراجرای آنها، چه کاری انجام دهند کـه موارد زیرامکان پذیرند:

۱- هماهنگ کننده پیش ازشروع کردن شیوه اجرا، قطع می شود (خراب میشـود) بنابراین شروع به انجام فرایند، باریکاوری خواهد کرد.

2- هماهنگ کننده، درحالیکه درحالـت آمـاده اسـت، قطـع میشـود. دراین مـورد هماهنـگ کننـده، فرمـان perepare (آمـادگی) را ارسـال کـرده اسـت. طـی ریکاوری، هماهنگ کننده فرایند اجرا را برای تـراکنش ازابتـدا بـا ارسـال یـک باردیگرپیام prepare (آمادگی) دوباره راه اندازی (ری اسـتارت) مـی کننـد. اگرشرکت کنندگان تراکنش را متوقف کرده باشند، می توانند به هماهنگ کننده اطلاع دهند. اگربلوکه شده باشند، آنها مـی تواننـد آراء اولیـه خـود را مجـددا ارسال کنند واجرای فرایند را ازسرگیرند.

3- هماهنگ کننده پس ازاینکه شرکت کنندگان را ازتصمیم سراسری وتراکنش متوقف شد، آگاه کرد، قطع شود. دراین مورد طی ریکاوری، نیازی به هیچ چیزنیست.

پروتکل های تکراری (برگشتی- روی هم)

درپایگاه داده توزیعی تکراری، هرمورد داده منطقی برخی نمونه های فیزیکی دارد. برای مثال حقوق یک کارمند (مورد داده منطقی) می تواند درسه بخش (کپی های فیزیکی) ذخیره شود. مسئله وموضوع این گونه ازپایگاه داده ها، حفظ مسائل هماهنگی درمیان کپی هاست. تطابقی ترین وسازگارترین معیارآن « تساوی یک کپی » است.

اگرآن پیدایی تکراری برقرارگردد، تراکنش ها عملیات های خواندن ونوشتن روی یک مورد داده منطقی انجام خواهندگرفت. پروتکل کنترل نسخه کپی پاسخگو به عملیات های نگاشتن روی x به عملیات روی کپی های فیزیکی ($x_1 - x_n$) خواهند بود. یک نمونه پروتکل کنترل نسخه کپی که قابلیت تسلسل یک کپی را اعمال می کند، پروتکل یک بارخواندن/ نوشتن همه (Rown) نامیده می شود. Rown همه خواندن را روی (Read X) برای خواندن روی یکی ازکپی های فیزیکی x_i Read X می نگارند. پروتکل Rowa، آسان وساده است. اما به آن نیازدارد که تمامی کپی های تمامی مورد داده های منطقی که توسط یک تراکنش به روز می شوند، برای تراکنش به منظورانسداد (قطع) دردسترس باشند. خطا و خرابی یک سایت (بخش) ممکن است یک تراکنش را

با کاهش دسترسی پایگاه داده، بلوکه کند. الگوریتم با بیشترین سازگاری وجوری می تواند ازیک چشم اندازنسبتا متفاوت ملاحظه گردد. این ضد ومخـالف آراء برابربـرای هرکپی وتراکنشی است که آن مورد داده منطقـی را کـه مـی توانـد بـا موفقیـت طبـق بیشینه آراء کامل شود، به روزکنند. برمبنای ایـن ایـده، یـک الگـوریتم رای گیـری بـا مبنای حد نصاب (اولیه) به رای را به هرکپی مورد داده کپی شده تشخیص می دهد. کپی هرعملیات باید یک حدنصاب خواندن (V_T) با یک حد نصاب نوشتن (V_W) برای خواندن با نوشتن یـک مـورد داده بدسـت آورد. اگریـک مـورد داده، مجمـوع V رای داشته باشد، حد نصاب ها باید ازقوانین زیرتبعیت کنند.

۱- $V_r + V_w \rangle V$ یک مورد داده طبق دوتراکنش همزمان، خواندنی ونوشتنی نیسـت، برای پرهیزازتداخل وتعارض خواندن- نوشتند

۲- $V_w \rangle \frac{V}{۲}$ (۲ عملیات نوشتن ازدوتـراکنش نمـی تواننـد همزمـان درمـورد داده ای یکسان روی دهند، برای پرهیزازتداخل وتعارض خواندن- نوشتن).

منابع

۱. *پایگاه داده* ؛ پدیدآورنده :محمدکریم سهرابی؛ ناشر :پوران پژوهش ؛ خرداد، ۱۳۸۸

۲. اصول طراحی *پایگاه داده* ها؛ پدیدآورنده :آبراهام سیلبرشاتس، عین الله ؛ جعفرنژادقمی(مترجم)؛ ناشر :علوم رایانه ؛ اسفند، ۱۳۸۷

۳. *پایگاه داده* ها؛ پدیدآورنده :حمیدرضا زهره وند، حامد زهره وند؛ ناشر : زهره وند مهر، ۱۳۸۸

٤. اصول طراحی *پایگاه داده* ها؛ پدیدآورنده :الهام میرزاکاظمی؛ ناشر :الهام میرزاکاظمی ؛ اسفند، ۱۳۸۸

فهرست مطالب