

اگر به ساز و کار شیرپوینت مایکروسافت دقت کنید، همه چیز را داخل دیتابیس ذخیره می‌کند (از اطلاعات رکوردها گرفته تا فایل‌ها و غیره). حال شاید این سؤال مطرح شود که برای ذخیره سازی فایل‌هایی با تعداد بیش از یک میلیون عدد، استفاده از دیتابیس مناسب است یا فایل سیستم متداول. برای پاسخ به این سؤال باید به نکات ذیل توجه داشت:

- هر نوع عملیاتی که بر روی فایل‌ها صورت گیرد، بستن، بازکردن و غیره، نیازمند اعمالی در سطح سیستم عامل است (برای مثال بررسی سطح دسترسی لازم برای انجام این کارها).
- هر گونه عملیاتی بر روی فایل‌ها نیازمند یک حداقل قفل گذاری بر روی آن‌ها است که این نیز مصرف CPU قابل توجهی را سبب خواهد شد.
- تمامی اعمال ذکر شده کل سرور و تمامی سرویس‌های در حال اجرا را تحت تاثیر قرار داده و بازدهی آن‌ها را کاهش می‌دهند.
- حتی سیستم عامل‌ها نیز از یک file system database جهت مدیریت اعمال خود استفاده می‌کنند اما این روش برای مدیریت میلیون‌ها و میلیارد‌ها فایل بهینه سازی نشده است.
- ذخیره سازی میلیون‌ها و میلیارد‌ها فایل به تدریج سبب ایجاد fragmentation قابل توجهی شده و این مورد نیز بر روی کارایی تاثیر منفی خواهد گذاشت (همچنین این مورد بر روی طول عمر تجهیزات ذخیره سازی داده‌ها تاثیر منفی دارند).
- تهیه پشتیبان و بازگرداندن میلیون‌ها فایل بسیار زمانگیر است (برای مثال جابجایی یک فایل یک مگابایتی بسیار سریعتر است از جابجایی 100 فایل 10 کیلوبایتی).
- مدیریت تغییرات و همچنین بررسی اینکه چه شخصی چه فایلی را قرار داده، حذف کرده یا تغییر داده است در حالت استفاده از file system مشکل است.
- به صورت پیش فرض عموماً مباحث replication و امثال آن توسط روش استفاده از file system خصوصاً با تعداد بالای فایل، پشتیبانی نمی‌شود.
- در حالت استفاده از file system، برنامه‌های وب باید دسترسی write بر روی یک سری پوشه داشته باشند که این مورد همیشه از دیدگاه امنیتی مساله ساز بوده و مشکل آفرین.
- کرش file system مساوی است با کرش سیستم عامل و بازگشت این‌ها زمان‌بر خواهد بود.

با توجه به این نکات استفاده از دیتابیس برای ذخیره سازی تعداد زیادی فایل، مزایای زیر را به همراه خواهد داشت:

- اکثر سیستم‌های دیتابیزی امروزی برای کار با حجم عظیمی از داده‌ها به حد بلوغ خود رسیده‌اند.
- هنگام استفاده از دیتابیس برای ذخیره سازی فایل‌ها دیگر سر و کار ما با میلیون‌ها فایل نخواهد بود و حداکثر چند فایل دیتابیس و ملحقات آن مانند لاگ فایل، کل سیستم را تشکیل می‌دهند.
- فایل‌های دیتابیس برای مثال SQL Server، همیشه توسط SQL Server در حالت باز قرار داشته و مباحث قفل‌گذاری بر روی فایل‌های دیتابیس و بررسی سطح دسترسی و غیره توسط سیستم عامل در این‌جا به حداقل خود می‌رسد.
- در این حالت بار سیستم عامل شما تنها سیستمی است که مشغول سرویس دهی اطلاعات دیتابیس‌های شما است.
- جستجوی فایل‌ها، حتی جستجو در محتوای این فایل‌های ذخیره شده در یک دیتابیس بسیار سریعتر از روش file system می‌باشد. امکان استفاده از کوئری‌های SQL انعطاف پذیری خاصی را به این سیستم‌ها خواهند داد (برای مثال قابلیت full text search مربوط به SQL server امکان جستجو بر روی رکوردهایی با محتوای pdf را نیز پس از انجام اندکی تنظیمات، دارا می‌باشد).
- هنگام کار با دیتابیس مباحث تراکنشی نقش بسیار حائز اهمیتی را بازی می‌کنند اما عموماً سیستم عامل‌ها در این زمینه نیازمند کار و برنامه نویسی قابل توجهی هستند (این قابلیت به ویندوز ویستا اضافه شده است).
- کرش یک دیتابیس عموماً سبب کرش سیستم عامل یا حتی کرش سایر دیتابیس‌های موجود نخواهد شد.
- امکان تهیه پشتیبان از دیتابیس‌ها و بازیابی آن‌ها ساده است. (حداقل از بازیابی میلیون‌ها فایل ساده‌تر است)
- امکانات replication به صورت پیش فرض در اکثر سیستم‌های دیتابیزی امروزی مهیا است.

- امکان ثبت وقایع و مدیریت اطلاعات افزوده شده به دیتابیس، از طریق نرم افزارهایی که برای این کار نوشته خواهند شد (یا حتی امکانات توکار این برنامه‌ها) از هر لحاظ نسبت به روش file system برتری دارد.
- امکانات سوئیچ کردن به دیتابیس دیگری در شبکه در صورت کرش یک نود، مهیا است و پیش بینی شده است.
- برای استفاده از یک دیتابیس توسط یک برنامه وب، نیازی به داشتن دسترسی write بر روی هیچ فولدري وجود ندارد که این خود یک مزیت امنیتی مهم است و همچنین امکان محدود کردن سطوح دسترسی به فایل‌های ذخیره شده در دیتابیس با برنامه‌های نوشته شده نیز ساده‌تر است. (البته در این‌جا مسلماً منظور از دیتابیس، دیتابیس Access نیست و SQL Server یا MySQL مد نظر هستند)

## نظرات خوانندگان

نویسنده: salarblog

تاریخ: ۱۳۸۸/۰۶/۱۰ ۲۲:۱۶:۳۴

مطلب جالبی بود.

این سوال برای من هم پیش اومده بود که چرا از SQL Server برای Team Foundation Server استفاده کردن، درحالی که این حجم زیاد تراکنش و فایل وجود داره. ولی باید اقرار کنم که خیلی سریع تر SourceControl2005 هست.

با این حال استفاده از دیتابیس مسئله قابلیت حمل رو پیش میاره.

نویسنده: macromediax

تاریخ: ۱۳۸۸/۰۶/۱۹ ۱۷:۱۲:۵۳

اما معایب استفاده از دیتابیس برای فایل‌ها رو ننوشتی که فکر میکنم در انتهای مقاله این مورد رو هم باید بررسی میکردی. مثلاً فرض کن یک سرویس نوشتی که فایل‌های آهنگ و عکس‌های کاربرانشو میگیره و باید ذخیره کنه و چندین بار پخش کنه یا نمایش بده. مسلماً نمایش هر عکس اگر بخواد از دیتابیس لود بشه، فشار زیادی رو روی dbms میذاره.. بخصوص وقتی تعداد عکس‌ها و درخواست نمایش بالاست..

نویسنده: وحید نصیری

تاریخ: ۱۳۸۸/۰۶/۱۹ ۱۷:۴۷:۴۰

مقاله کلا در این مورد بود که اگر تعداد عکس‌ها بالا باشد روش سنتی ذخیره سازی روی هارد به دلایلی که ذکر شده مشکل زا خواهد شد و اس کیوال سرور اینقدر توانمند و پخته است که بتواند این مساله را حل و فصل کند. (نمونه استفاده در مقیاس سازمانی از آن شیرپوینت است که همه اطلاعات را داخل دیتابیس قرار داده و سپس لود می‌کند) یا برای نمونه YouTube نیز تمام ویدیوهای خودش را درون دیتابیس ذخیره کرده و صد البته برای مقیاس پذیری آن روش‌های کش کردن اطلاعات پیشرفته‌ای را نیز توسعه داده‌اند. شیرپوینت هم برای این منظور چیزی به نام BlobCache دارد.

نویسنده: Saeed

تاریخ: ۱۳۸۹/۰۳/۲۲ ۱۷:۰۳:۴۸

خیلی ممنون از مطلب خوبتون.

خیلی وقت بود که به دنبال جواب این سوال بودم که برای ذخیره کردن فایلها در سمت سرور از کدوم روش استفاده کنم. البته به این نتیجه رسیده بودم که فایلها رو در بانک اطلاعاتی ذخیره کنم. بعد از انجام این کار متوجه شدم که برای ذخیره کردن حدود 20 مگابایت عکس در سمت سرور، به اندازه فایلهای بانک اطلاعاتی من در حدود 80 مگابایت افزوده می‌شود. که این امر من رو برای ذخیره و نگهداری حجم و تعداد زیاد فایل نگران کرده بود.