BLM 3780 HW2

Ömer Diner

20011017

omer.diner@std.yildiz.edu.tr

Veritabanı Yönetim Sistemlerinde Kurtarma

VTYS'ler her an yüzlerce, binlerce işlemin gerçekleştiği karmaşık yapılardır. Bir VTYS'nin dayanıklılığı ve sağlamlığı, mimarisine ve altında yatan donanım ve sistem yazılımına bağlıdır. İşlemler sırasında arızalanır veya çökerse, sistemin kayıp verileri çeşitli yöntemler kullanarak geri kurtarabilmesi beklenir.

Veritabanında Oluşan ve Kurtarma Gerektirebilecek Arıza Türleri

Transaction Hatası: Bir işlem yürütülemediğinde veya devam ettirilemeyeceği bir noktaya ulaştığında, işlem başarısız olur. VTYS işlemi gerçekleştiremediğinde transactionu sonlandırmalı ve stabil halini korumalıdır.

Disk Arızası: Disk arızaları, sabit disklerin ve depolama sürücülerinin işlevlerinde hata nedeniyle meydana gelir. Disk arızasının çeşitli nedenleri bad sector oluşumu, erişilemezlik ve fiziksel hasarlardır.

Sistem Çökmesi: Elektrik kesintileri, donanım ve yazılım arızaları; işletim sisteminin kilitlenmesi gibi sistem çökmelerine neden olabilir. Nonvolatile bellek(ssd, hdd vb.) sistem çökmesinden etkilenmez.

Kurtarma ve Atomiklik

Bir VTYS bir çökmeden kurtulduğunda aşağıdakileri sağlamalıdır:

- Yürütülmekte olan tüm transactionların durumları kontrol edilmeli.
- Bir transaction bir işlemin ortasında olabilir; VTYS bu durumda transactionun atomikliğini sağlamalıdır.
- Transactionun şimdi tamamlanıp tamamlanamayacağını ya da geri alınması gerekip gerekmediğini kontrol etmelidir.
- Hiçbir transactionun VTYS'yi tutarsız bir durumda bırakmasına izin verilmemeli.

Veritabanlarında genel olarak dört farklı kurtarma tekniği uygulanır.

Mirroring

Veritabanının iki tam kopyası, farklı kararlı depolama aygıtlarında online olarak tutulur. Bu yöntem çoğunlukla kesintisiz, hataya dayanıklı operasyonlar gerektiren sistemlerde kullanılır. Veritabanı aynası(mirror), birincil veritabanının arızalanması durumunda kullanılabilecek veritabanının tam bir yedeğidir. Birincil veritabanındaki işlemler ve değişiklikler doğrudan ikinci veritabanına da aktarılır ve hemen işlenir, böylece her zaman güncel olan bir veritabanımız daha olur ve "etkin" yedek olarak kullanılabilir.

Yedekleme

Yedekler sistemde büyük kayıp oluştuğu durumlarda kurtarma için etkilidirler. İki tür yedekleme vardır:
1. Anlık Yedekleme: Anlık Yedekleme bir diskette, sabit diskte veya manyetik bantlarda tutulur. Bunlar, sistem arızası, disk çökmeleri, ağ arızası gibi birincil veritabanında teknik bir arıza meydana geldiğinde kullanışlı olur.

2. Arşiv Yedekleme: Bu yedekler manyetik bant, CD-ROM'lar gibi yığın depolama cihazlarında tutulur. Yangın, deprem, sel gibi felaketlerden sonra verileri kurtarmak için kullanışlıdırlar. Bu yedekler, sistemin çalıştığı yerden farklı bir yerde tutulmalıdır.

Loglar Kullanılarak Kurtarma

Bu yöntemde takip edilen adımlar aşağıdaki gibidir:
- Kurtarma sistemi, logları sondan checkpointe kadar geriye doğru okur.
- Bir undo listesi ve bir redo listesi olmak üzere iki liste tutar.
- Kurtarma sistemi <Tn, Start> ve <Tn, Commit> veya sadece <Tn, Commit> içeren bir kayıt görürse, işlemi redo listesine koyar.
- Kurtarma sistemi <Tn, Start> içeren bir kayıt görürse ancak commit veya abort logu bulunamazsa, işlemi undo listesine koyar. Undo listesindeki tüm işlemler sonra geri alınır ve logları kaldırılır. Redo listesindeki tüm transactionlar ve bunların önceki logları kaldırılır ve logları tekrar kaydedilmeden önce yeniden yapılır.

Shadow Paging

Gölge sayfalama, sayfaların yerinde güncellenmesini önlemek için kullanılan bir copy-on-write tekniğidir. Bunun yerine, bir sayfa değiştirileceği zaman bir gölge sayfa tahsis edilir. Gölge sayfanın hiçbir referansı olmadığından (diskteki diğer sayfalardan), tutarlılık kısıtlamaları vb. endişesi olmadan serbestçe değiştirilebilir. Sayfa dayanıklı hale gelmeye hazır olduğunda, orijinal sayfaya referans veren tüm sayfalar, bunun yerine yeni yedek sayfaya referans verecek şekilde güncellenir. Sayfa yalnızca hazır olduğunda "etkinleştirildiği" için atomiktir.

Eşzamanlı Transactionlarda Kurtarma

Birden fazla transaction paralel olarak yürütüldüğünde, loglar birbirine eklenir. Kurtarma sırasında, kurtarma sisteminin tüm logları geri takip etmesi ve ardından kurtarmaya başlaması zor olacaktır. Bu durumu basitleştirmek ve hızlandırmak için çoğu modern VTYS checkpoint kavramını kullanır.

Checkpoint

Logların gerçek zamanlı olarak tutulması ve muhafaza edilmesi, sistemde mevcut olan bellek alanını doldurabilir. Zaman geçtikçe, log dosyası hiç işlenemeyecek kadar büyüyebilir. Checkpoint, önceki tüm logların sistemden kaldırıldığı ve bir depolama diskinde kalıcı olarak saklandığı bir mekanizmadır. Checkpoint, VTYS'nin daha önce tutarlı bir durumda olduğu ve tüm işlemlerin gerçekleştirildiği bir noktayı bildirir. Bu noktaya gelindiğinde sistemin tutarlı olduğu bilinir, bu noktadan sonraki işlemler için loglar takip edilir.

Güncel Konular

Günümüzde popülerlik kazanan ve tartışılan bir yöntem ise bulut teknolojilerini kullanarak yedekleme ve kurtarma işlemlerini yapmaktır.

Bulut tabanlı yedekleme ve kurtarma kolay ölçeklenebilirlik, kullanılabilirlik, maliyet verimliliği ve güvenlik gibi avantajlar sunabilir. Bulut tabanlı yedekleme ve kurtarma ayrıca farklı konumlar ve platformlar arasında daha hızlı ve kolay veri geçişi, replikasyonu ve senkronizasyonu sağlayabilir. Bununla birlikte, bulut tabanlı yedekleme ve kurtarma bant genişliği sınırlamaları, veri egemenliği, uyumluluk ve büyük şirketlere dolayısıyla yurtdışına bağımlılık gibi bazı zorlukları da beraberinde getirir. Bu nedenle, sistem yöneticilerinin ve şirketlerin VTYS yedekleme ve kurtarma ihtiyaçları için bulutu kullanmanın artılarını ve eksilerini düşünmeleri gerekir.

İleride yaygınlaşabileceğini düşündüğüm bir başka yöntem ise bu işlemlerde popülerliği ve işlevi gittikçe artan yapay zeka ve makine öğrenmesini kullanmak.

Yapay zeka ve makine öğrenimi, sistem yöneticilerinin veri özelliklerine, kullanım modellerine ve performans metriklerine dayalı olarak yedekleme ve kurtarma politikalarını, programlarını ve yöntemlerini optimize etmelerine yardımcı olabilir. Yapay zeka ve makine öğrenimi ayrıca olası arızaları veya tehditleri tespit edip önlemeye ve bu durumlardan hızlı bir şekilde kurtulmaya yardımcı olabilir. Ayrıca yapay zeka yedekleme ve kurtarma sonuçlarını iyileştirmek ve riskleri , maliyetleri azaltmak için içgörüler ve öneriler sağlayabilir. Mesela VTYS'e entegre edilmiş bir chatbot sistemi veritabanı yöneticilerine kurtarma ve yedekleme işlemleri için öneriler verip, danışmanlık yapabilir.

Şimdiye kadar anlatılan yöntemler ve problemler ağırlıklı olarak ilişkisel veritabanları için geçerli olan durumlar ve bunları çözmek için geliştirilen çözümlerdi. Bu kısımda da NoSQL veritabanlarında kurtarma ve yedekleme işlemlerinde yaşanan sıkıntı ve getirilen çözüm denemeleri ele alınacak.

NoSQL Veritabanlarında Kurtarma ve Yedeklemede Sıkıntılar

1) Dağıtık Mimari: Yüksek kullanılabilirlik ve ölçeklenebilirlik elde etmek için NoSQL veritabanlarının birkaç düğüm ve veri merkezi üzerinde dağıtılması amaçlanmıştır. Ancak bu mimari, yedekleme ve kurtarma işlemleri sırasında veri tutarlılığı yönetimini zorlaştırır. Verilerin tüm düğümlerden yedeklendiğinden ve tutarsızlık olmadan geri yüklendiğinden emin olmak zordur.

2) Veri Modeli Çeşitliliği: NoSQL veritabanları belge, anahtar-değer, sütun ailesi ve grafik veritabanları dahil olmak üzere çeşitli veri modellerini destekler. Her modelin kendine özgü veri yapıları ve sorgu dilleri vardır, bu da tüm bu modelleri destekleyen birleşik bir yedekleme ve kurtarma stratejisi oluşturmayı zorlaştırır.
3) Standardizasyon Eksikliği: Yedekleme ve kurtarma için köklü standartlara sahip olan geleneksel ilişkisel veritabanlarının aksine, NoSQL araçlar, formatlar ve yöntemler açısından standarttan yoksundur. Standardizasyon eksikliği, kapsamlı yedekleme ve kurtarma çözümlerinin oluşturulmasını zor hale getirmektedir.

4) Yedekleme Tutarlılığı: NoSQL veritabanlarında veri kaybını veya çoğaltılmasını önlerken uzak düğümler arasında tutarlı yedeklemeler sağlamak zordur. Sistem çalışmaya devam ederken düğümler arasında yedekleme sürecinin koordinasyonu, titiz bir senkronizasyon gerektirir. NoSQL veritabanları genellikle veritabanlarını kesin bir geçmiş durumuna geri yüklemek için kritik öneme sahip olan zaman içindeki bir noktaya dönerek kurtarma için gereken destekten yoksundur.

Bu sorunları aşmak için geliştirilen yöntemler:

1) Parçalı ve Dağıtılmış Yedeklemeler
Artımlı ve dağıtılmış yedekleme çözümlerini kullanarak veritabanı performansı fazla düşürülmeden dağınık düğümler arasında tutarlı yedeklemeler sağlanabilir. Bu, önceki yedeklemedeki değişiklikleri tespit etmeyi ve güncellenmiş verileri kopyalayıp kaydetmeyi gerektirir.

2) Veri Modeline Duyarlı Yedekleme çözümleri
Çeşitli veri modellerinin inceliklerini kavrayan ve prosedürlerini buna göre değiştiren yedekleme çözümleri oluşturmak verimliliği ve doğruluğu artırır. Belirli veri modellerini etkin bir şekilde yedeklemek için özelleştirilmiş komut dosyaları veya üçüncü taraf araçlar kullanılabilir.

3) Otomasyon ve Düzenleme
Otomatik yedekleme ve kurtarma yöntemleri, düzenleme frameworkleriyle birlikte uzak düğümler arasında yedekleme görevi yönetimini basitleştirir. Bu yöntem, düzenli ve güvenilir yedeklemeler sunarken manuel müdahale ihtiyacını azaltır.