

Gelişmiş Hareket İşleme

Kerem ERZURUMLU
A0064552

İçindekiler

1. Uzak Yedekleme Sistemleri	2
a. Hatanın Yakalanması.....	3
b. Aktarımın Kontrolü	3
c. Kurtarma Zamanı	3
d. Onaylama Zamanı	3
2. Hareket İşleme Monitörleri.....	3
2.1 Hİ-Monitör Mimarileri	4
2.2 Hİ-Monitör Kullanarak Uygulama Eşgüdümlemesi.....	6
3. Yüksek Başarılı Hareket Sistemleri	6
3.1 Ana Bellek Veritabanları	6
3.2 Grup Onaylama	7
4. Uzun Süreli Hareketler	7
5. Gerçek Zamanlı Hareket Sistemleri	7

Gelişmiş Hareket İşleme

Genel olarak 'Basit Hareket İşleme'de veri tabanlarının genenekselsel ACID özellikleri sağlamaya yöneliktir. Fakat 'Basit Hareket İşleme'nin öngürmediği bazı durumlar söz konusudur ve bu durumlar veri tabanı özelliklerini tehdit etmektedir. Bu nedendir ki 'Basit Hareket İşleme'nin yanı sıra bir de 'Gelişmiş Hareket İşleme' söz konusudur.

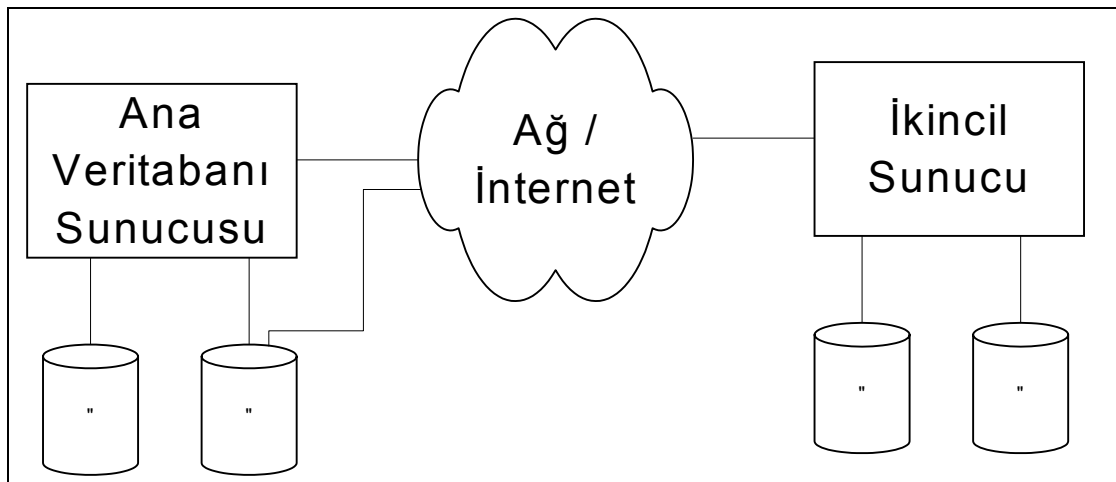
Bu inceleme yazısında basit sistemlerinde ötesindeki "Uzak Yedekleme Sistemleri", "Hareket İzleme Monitörleri", "Yüksek Başarımlı Hareket Sistemleri", "Uzun Soluklu Hareketler" ve "Gerçek Zamanlı Hareket Sistemleri" başlıkları altında 'Gelişmiş Hareket İşleme'yi inceleyeceğiz.

1. Uzak Yedekleme Sistemleri

Genenekselsel hareket işleme sistemleri merkezileştirilmiş ve istemci-sunucu mimarisi ile çalışan mekanizmalardır. Bu ve benzeri sistemler deprem, sel ve yangın gibi sunucu bilgisayar sistemlerine fiziksel hasar verebilecek etmenlere karşı aşırı zayıftır. Bu nedenle bu etmenlerden etkilenmeyen, "yüksek bulunurluk" sağlayan hareket yöntemlerinin gerekliliği kaçınılmazdır.

Yüksek bulunurluk, verilerin ve hareketlerin kopyalarının ayrı bir konumda tutulduğu dağıtık bir veri tabanı kullanılarak sağlanabilir. Sunucuların senkronizasyonu için "iki aşamalı onaylama" (two-phase commit) kullanabilir. Böylece sunucu makinalardan biri ulaşılamaz durumda olsa dahi diğer sunucu üzerinden işlemler devam ettirilebilir. Fakat yalnızca bir sunucuda işletilen sorgular veri bütünlüğünde bir tutarsızlık meydana getirme riskini taşır. Bunun haricinde iki aşamalı onaylama'nın başlı başına yüksek bir maliyeti vardır.

Bu karışık ve maliyetli duruma çözüm olabilecek bir diğer durum ise tüm hareketlerin "ana sunucu" da işletilmesidir. Fakat bu "ana sunucu"nun bir "uzak yedek sunucusu" olacak ve ana sunucudaki tüm veriler bu yedek sunucusunda kopyalanacaktır. Bahsi geçen yedek sunuculara ekseriyetle "İkincil Sunucu" adı verilmektedir. İkincil sunucu, ana sunucu ile ana sunucu üzerindeki her güncelleme işlemi sonucunda senkronize edilmek zorundadır. Genelde senkronizasyon ana sunucudan son senkronizasyon işleminden sonra gerçekleşen işlemlerin kayıtlarının gönderilmesi ile olur. Tavsiye edilen sistem İkincil Sunucuların mümkün olduğu kadar ana sunucudan uzak konumlandırılmasıdır ki ana sunucuya zarar veren bir faciadan yedek sistemler etkilenmesin. Şekil 1 örnek bir yedekleme mekanizmasını göstermektedir.



Şekil 1: Genel Bir Uzak Yedekleme Sistemi Şeması

Ana sunucuda bir arıza meydana geldiğinde ve istemlere cevap veremediğinde İkincil Sunucu devreye girer ve istemlere cevap vermeye başlar. Bu arada kendi içerisinde ayrı bir kayıt tutmaya başlar. Bu kayıt ana sunucunun koptuğu andan itibaren gerçekleşen kayıtlardır. Ana

Gelişmiş Hareket İşleme

sunucu tekrar çevirim-içi durumuna geldiğinde otomatik olarak ikincil sunucuya bakar ve aralarında bir fark mevcut ise bu farkı alarak iki sunucunun senkronize olmasını sağlar. Böylece veri tutarlılığı sağlanmış olur.

Tabiki bu mekanizmada da çok da az olsa bile veri kaybetme riski mevcuttur. Kaybolma riski taşıyan veriler henüz işlemi tamamlanmamış hareketlerdir. Uzak yedekleme mekanizmalı bir sistemin başarımı iki aşamalı onaylama kullanan bir sisteme oranla daha iyidir.

Bir uzak yedekleme mekanizmasının kurulması/tasarlanması esnasında takip eden maddelerin göz önünde bulundurulması düşünülmeli gerekmektedir;

- a. Hatanın Yakalanması:** Ayrık sistemlerde de önemli olduğu üzere ikincil sistemin ana sunucunun arıza durumunu yakalayabilmesi gerekmektedir. Tabi ki ikincil sunucu hatlarda meydana gelen bir hata nedeni ile ana sunucuyu arızalı sanabilir ve işlemlere devam etmeye kalkabilir. Bu veri tutarlılığını riske atacağı için genelde ana sunucu ile ikincil sunucu bir telefon hattı ile birbirine bağlanır. Böylece bu risk en aza indirilmiş olur.
- b. Aktarımın Kontrolü:** Bir hata durumunun giderilmesinden sonra uygulanabilecek iki seçenek mevcuttur. Birincisi ana sunucu görevini üstlenen ikincil sunucunun bu işe devam etmesi ve hatası düzeltilen ana sunucunun yedek sunucu olarak çalışması, ikincisi ise ana sunucunun ilk görevini alması. Kesintisiz hizmet bağlamında iki yöntem arasında da bir fark yoktur. Buradaki tercih tamamen makinaların özelliklerine ve sistemi kuran kişilerin tercihlerine kalmıştır.
- c. Kurtarma Zamanı:** Yedek sistem işlemleri üzerine aldığı ana sunucu uzun süre kendine gelemeyen yada hatası düzeltilemez ise yedek sunucudaki kurtarma kayıtları büyür ve geri kurtarma işlemi bu kayıtlar büyüdükçe daha fazla zaman isteyen bir işlemdir. Bu nedenle ikincil sunucu belli bir noktadan sonra bazı verileri "tekrar gerçekleştirme kayıtları"ndan çıkartabilir. Şüphesizki bu çıkarttığı kayıtlar daha sonradan modifiye edilmiş ve geçerliliğini yitirmiş kayıtlar olmak zorundadır. Böylece gereksiz kayıtlar ana sunucuya hiç iletilmemiş olur.
- d. Onaylama Zamanı:** Bir hareketin kayıtları ikincil sunucuya ulaşmadan "onaylanmış" sayılmaz. Bu kullanıcılar/istemciler açısından bir gecikme olarak görülür. Bu nedenle üç yöntemden biri onaylama zamanı olarak seçilir;
 - a. Tek Güvenli:** Onaylanan bir hareket yalnızca ana sunucuda sabit bir veri alanına yazılır. Bu mekanizmada belirli aralıklarla hareketler ikincil sunucuya iletildiği için bazı hareketlerin kaybolma ihtimali mevcuttur.
 - b. Çift Aşırı Güvenli:** Onaylanan bir hareket hem ana sunucuda hemde ikincil sunucuda sabit bir veri alanına yazılır. Bu mekanizma kullanıcıyı en çok bekleten ve en çok vakit kaybettiren yöntemdir.
 - c. Çift Güvenli:** Onaylanan bir hareket iki sunucuda geçerli hale getirilir fakat disk alanlarına hemen taşınmaz.

Bunların haricinde sunucu bilgisayar sisteminden kaynaklanabilecek diğer arızalara karşı da bazı önlemler mevcuttur. Örneğin ana sunucu bilgisayarda birden çok işlemci bulunabilir ve işlemciler birbirini yedekler. Aynı şekilde fiziksel verileri tutan disk sistemleri de RAID teknolojilerinden yararlanabilir ve sistemin donanım arızası nedeni ile devre dışı kalma ihtimalini en aza indirir.

2. Hareket İşleme Monitörleri

Geleneksel veritabanları sistemleri çoğunlukla merkeziliyetçi yapıda görünür. Yani verilerin bir çok veritabanına, dosyalara yada uygulamalara dağıldığı pek görülmemektedir. Fakat gerçek hayatta yaşananlar ve gereksinimler yukarıdaki gereksinimlere ihtiyaç duymaktadır. Bu süreçler

Gelişmiş Hareket İşleme

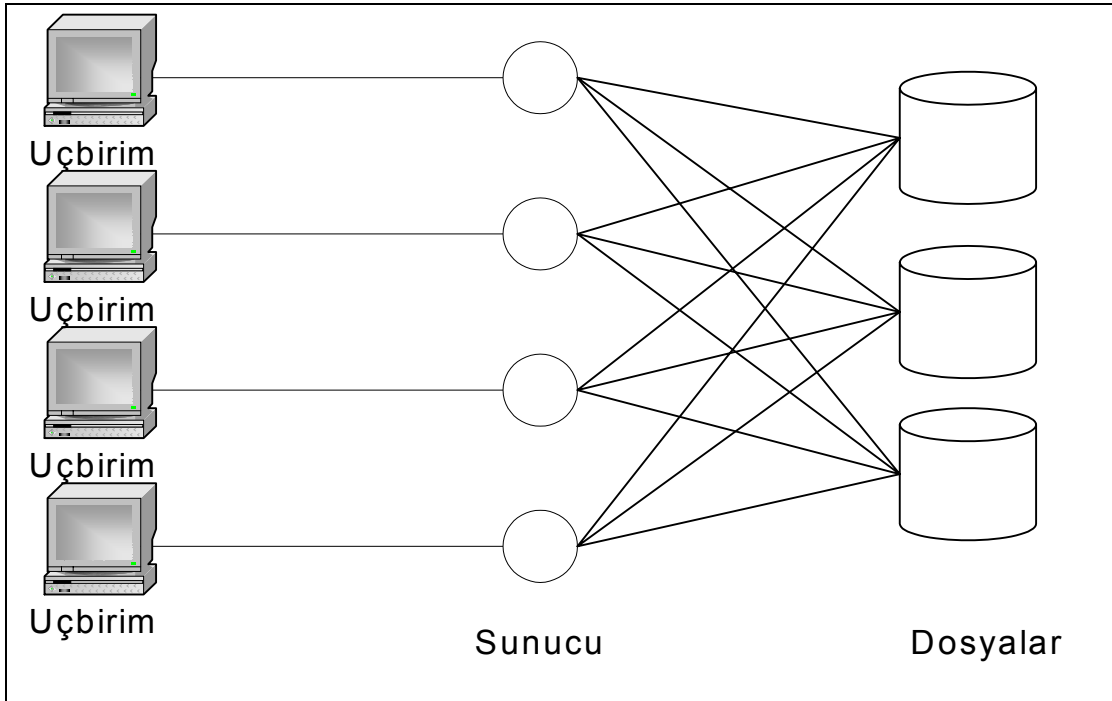
doğal olarak kontrol mekanizmalarının büyümesine neden olur ki bu da işlemlerin yavaşlamasını beraberinde getirir. Bu yavaşlık handikapını ortadan kaldırmak için ara bir hareketleri izleme ve kontrol etme mekanizması gerekmektedir. "Hareket İşleme Monitörleri" bu izleme ve kontrol etme işlemlerini gerçekleştiren mekanizmalardır.

Hareket işleme monitörleri 1970 ve 1980'li yıllarda bir çok uçbirime hizmet vermeye başlanması ile geliştirilmiştir.

2.1 Hİ-Monitör Mimarileri

Günümüz bilgisayar sistemlerinde genelde kullanıcılar bir sisteme girer ve o sistemde bir yada daha fazla işlemi gerçekleştirirler. Havayolu bilet gişeleri gibi binlerce uçbirimi olan bir sistemde tüm uçbirimlerin bir sunucuya bağlanması hali ile hem terminaller açısından hemde sunucu açısından çok yorucu bir işlem olacaktır.

Her uçbirimin sunucu sistemine bağlanması güvenlik nedenleri ile tercih edilmemektedir. Bunun yerine tercih edilen yöntem sunucuda her uçbirim için bir görevin çalışması ve bu uçbirim için gerekli olan işlemleri yerine getirmesidir. Şekil 2'de de gösterilmiş olan bu mimarinin yüksek bellek gereksinimi ve yüksek işlemci gereksinimi nedeni ile bazı problemleri mevcuttur.

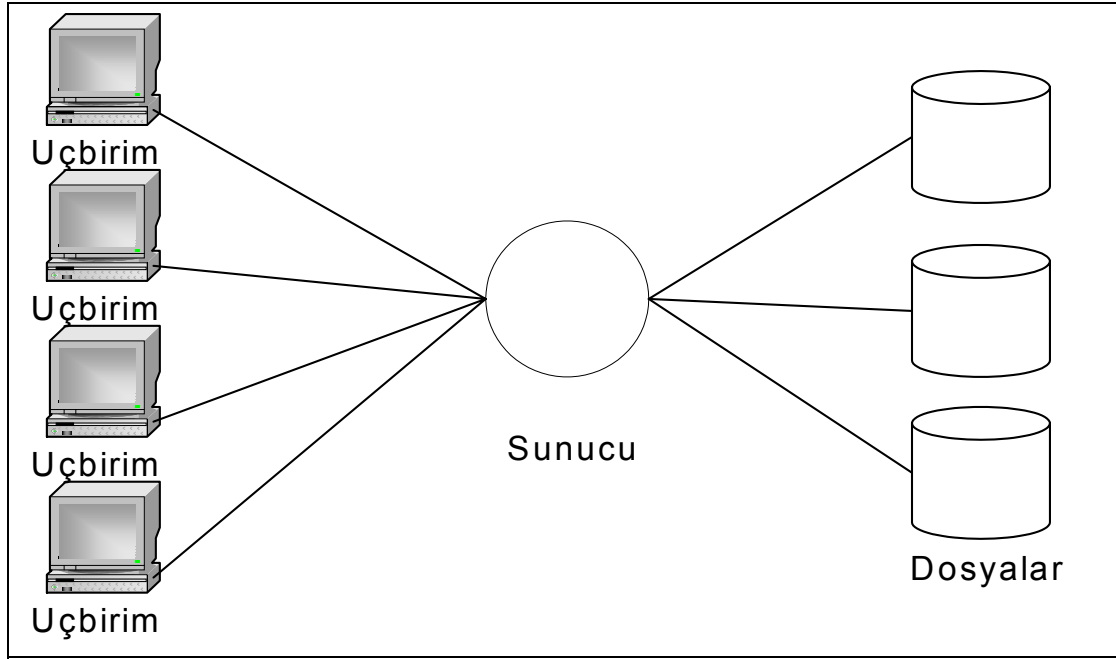


Şekil 2: İstemci Başına Görev Mimarisi

- İşlem öncesi bellek gereksinimleri çok fazladır. Çünkü her görev kendi yerel değişkenlerini ve verilerini tutmaktadır.
- İşlemci zamanının bölünmesi söz konusudur. Görevler arası anahtarlama işlemleri birkaç yüz mikro saniye tutmasına rağmen çok uçbirimli mimarilerde işlemci işlemlerden çok anahtarlama işleri ile uğraşmaktadır.

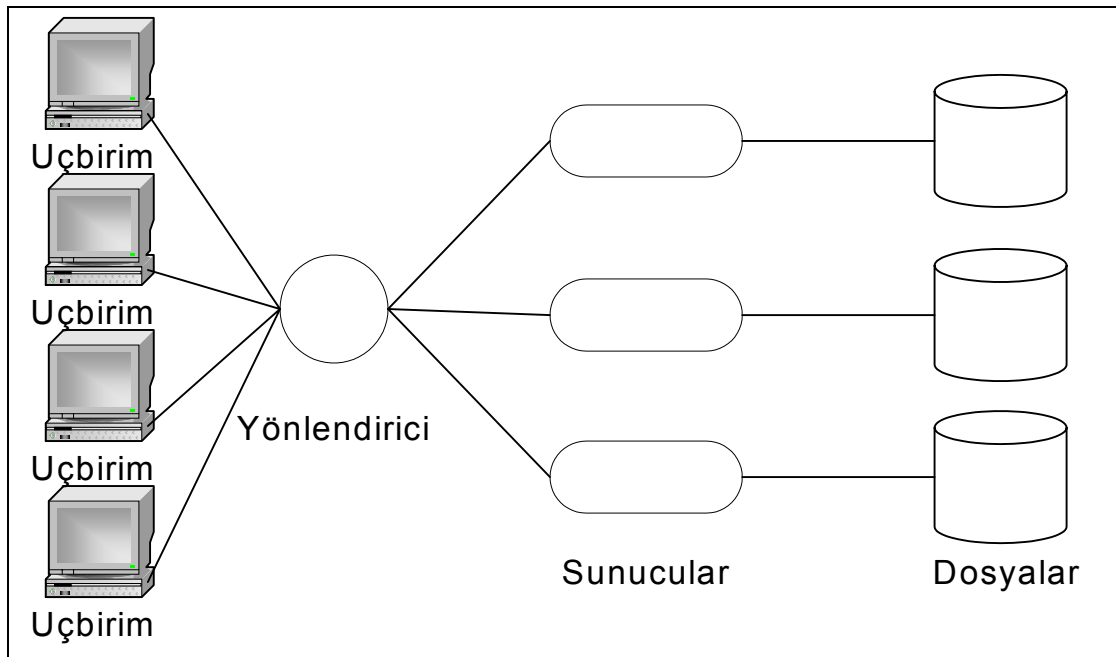
Bu mimari ile işlemci ve bellek miktarının çok kısıtlı olduğu 1970 ve 1980'li yıllarda 100'den fazla uçbirime aynı anda hizmet vermek mümkün değildi.

Bu problemlerden kaçmak amacı ile IBM Hareket İzleme Monitörleri kavramını ortaya çıkardı. Bu mantıkta sunucuda tek bir görev çalışıyor ve tüm istemcilerin işlemlerini tek başına gerçekleştirir. İstemcilerin çok beklemesini önlemek amacı ile sunucu çoklu kullanım yapıyor ve gerekli durumlarda sunucu programı çoğaltabilme kapasitesine sahiptir. Şekil 3 bu mimarinin genel bir görünümünü göstermektedir.



Şekil 3: Tek Görev Mimarisi

Tek Görev Mimarisi'nde bellek ve işlemci kullanımı aşırı olmamasına rağmen bu sistemler birden farklı sunucuda aynı anda çalışmayacağından dolayı paralel yada dağıtık veritabanları için uygun değildir. Bu duruma bir çözüm, arka planda birçok veritabanı sunucusu çalışırken, istemcilerin istemlerini yönlendiren bir ara katman aracılığı ile istemlerin farklı veritabanı sunucularına dağıtılması olarak öngörülmüştür. Bu mimariye "çok sunucu, tek yönlendirici" mimarisi adı verilir. Şekil 4'de şemalaştırılmış olan bu mimari sunucunun nasıl çalıştığından bağımsız olmakla birlikte gelende çoklu görevli sunucuları tercih etmekte ve aynı sunucu üzerinde birden çok uygulamanın sunumuna olanak sağlamaktadır.



Şekil 4: Çok Sunucu, Tek Yönlendirici Mimarisi

Günümüzde ise daha çok tercih edilen teknoloji 3 katmanın tamamında darboğaz olmaktan kurtarılmasına dayanmaktadır. Bu nedenledir ki "çok sunucu, tek yönlendirici" mimarisinin bir sonraki evrim noktası "çok sunucu, çok yönlendirici" olmuştur. Böylece sistemin hizmet sunabileceği istemci sayısı katlanarak çoğalmaktadır.

2.2 Hİ-Monitör Kullanarak Uygulama Eşgüdümlemesi

Günümüz teknolojisinde ve koşullarında uygulamalar birden çok veritabanı sistemi ile iletişim kurmak durumundadır. Tabiki uygulamalar yalnızca veritabanları sistemleri ile değil, aynı zamanda başka uygulamalarla iletişim kurmak durumunda da olabilirler. Şüphesizki bu çoklu ve karışık ağ iletişimi veritabanlarının ACID özelliklerini riske atan faktörler içermektedirler.

Modern Hİ monitörleri "kaynak yönetimi" temeline dayanmaktadır. Kaynak yönetimi arayüzü X/Açık Dağıtık Hareket İşleme standartlarıncı tanımlanmıştır. Bir çok veritabanı Açık Dağıtık Hareket İşleme standartlarını tanımakta ve desteklemektedir.

3. Yüksek Başarılı Hareket Sistemleri

Saniyede yüzler yada binler düzeyinde yüksek oranlı hareket işlemek istediğimizde donanımımızın da yüksek başarılı olması gereksinimi kaçınılmazdır. Paralleştirilebilme tek başına istenilen başarıyı sağlamaktan uzaktır. Disk erişimleri işlemci yada veriyolu hızlarındaki gelişim kadar hızlı olamamaktadır. Günümüzde dahi basit bir disk erişimi en hızlı diskler için bile 10 milisaniye gecikme ile gerçekleşebilmektedir. Bu neden ile disk erişimleri veritabanı sistemlerinin "en zayıf halkası"dır. Paralleştirme ise aynı data üzerinde paralelleştirme olması durumunda veri bütünlüğüne zarar verebilecek bir işlem olduğundan dikkatli tasarlanması gerekmektedir.

Bu dezavantajları ortadan kaldırmak için iki mimari söz konusudur. Birincisi "Ana Bellek Veritabanları", ikincisi ise "Grup Onaylama"dır.

3.1 Ana Bellek Veritabanları

Genellikle veritabanları sistemlerinin performansları verileri sabit disklerden ne kadar sürede okuyabildikleri ile sınırlıdır. Çünkü günümüz bilgisayarlarının en yavaş etmeni sabit disklerdir. Bu nedenle daha hızlı veritabanları amacı ile çalışmalar yaptığımızda sabit disklerde bulunan tüm verilerin ana bellek ünitelerinde tutulması hız açısından başarıyı en üst düzeyde artıracaktır. Zira ana bellek üniteleri sıradüzensel erişim değil doğrudan erişim yapmaktadırlar. Ve hızları sabit diskler ile karşılaştırmayacak kadar hızlıdır. Günümüzün koşullarındaki bir 64-bit tabanlı bir sunucu 14 Gigabayt'a kadar ana belleği desteklemektedir.

Tüm veritabanının bellekte tutuluyor olması ile sabit disklerdeki yalnızca 10 milisaniyelik gecikmeyi ortadan kaldırmaz. Aynı zamanda saniyede tamamlanan işlem sayısı adına da büyük katkılarda bulunur. Bu nedenledir ki "gerçek zamanlı veritabanları" ana bellek üzerinde tutulmak zorundadır.

Tüm bunlara rağmen ana bellek veritabanları da bazı noktalarda disk erişim hızlarına mahkumdurlar. Bu noktalar;

- Hareket onaylanmadan önce kayıt dosyaları geçerli ve kalıcı bir saklama ortamına kaydedilmelidir. Disk erişiminin bir alt limit olmaması için tercihen pil destekli ana bellekler kullanılır. Bir diğer seçenek ise daha sonra açıklayacak olduğumuz "grup onaylama" kullanmaktır.
- Sunucu bilgisayar uzun süre açık kalıyor ve işlemler sürekli devam ediyorsa, güncellenmiş bilgiler sabit diske geçirilemeden bellek sıkıntısı yaşanma ihtimali mevcuttur. Sistem bu durumda ilgili bellek bölmesinin sabit diske geçirilmesini beklemek durumunda kalacaktır.

Gelişmiş Hareket İşleme

- Sistem arızalarında bellek pil destekli değilse tüm veriler kaybolacaktır. Ayrıca sistem'in sıfırdan yeniden açılışı ve tüm veritabanının ana beleğe yüklenmesi vakit alacaktır.

Bu handikapların yanı sıra "ana bellek veritabanları" aşağıda listelenen eniyileştirmelere de olanak sağlar;

- Ana bellek maliyetlerinin disk maliyetlerinden daha pahalı olması nedeni ile veri yapıları daha dikkatli tasarlanmalıdır. Fakat ana belleğin bir avantajı farklı sayfalar arasında "bellek işaretçileri" (memory pointers) kullanılabileceği için veri tekrarını indirger.
- Sorgu sonuçlarının alınması işlemi eniyileştirilmeli ve mümkün olan en az bellek ile sonuçlara ulaşabilmelidir.
- Disk erişim alt limiti ortadan kaldırıldığında yeni bir alt sınır ile karşılaşılır; kilitlemeler. Bu nedenle uygulamalardaki kilitleme mekanizmaları dikkatlice tasarlanmalıdır.

3.2 Grup Onaylama

Onaylama işlemlerinin belli bir grubun tamamını içermesi ve kapsamasıdır. Böylece her hareket için ayrı ayrı onaylama gerçekleştirmektense bir grup hareket için toplu onaylama gerçekleşir ve bu grup boylarına ve hareket türüne bağlı olmakla birlikte %10 civarında bir başarımlı artışı sağlar.

4. Uzun Süreli Hareketler

Hareketlerin ilk tasarlandığı zamanlar, hareketlerin yalnızca veri işlemlerinden meydana geleceği ve kısa sürede tamamlanacağı düşünülmüştü. Fakat insan iletişimi devreye girdiğinde hareketler tanımlandığı gibi kısa süreli olarak kalmadı ve hareketlerin süreleri uzadı. Uzun süreli hareketler aşağıdaki karakteristik özellikleri göstermektedir;

- **Uzun Sürelidir:** İnsan ile iletişim söz konusu olduğunda insanların bilgisayarlar kadar hızlı olmaması sonucu hareketler bilgisayar ile gerçekleştirilebilecek bir işlemde daha uzun sürerler.
- **Onaylanmamış Veri:** Hareketlerin uzun süreli olması dolayısı ile insanlar onaylanmamış bir veri üzerinde işlemlerini yaparlar. Başka kişiler ise bahsi geçen aynı veri ile bir işlem yapacaklarında verinin eski onaylı hali mi yoksa onaylanmamış olan sürümünü mü kullanacakları belirsizdir. Çünkü onaylanmamış olan veriden vazgeçilebilir.
- **Altgörevler:** Kullanıcı yapacağı işlemleri bir çok altgörev oluşturarak yapabilir. Çalışma esnasında ise herhangi bir altgörevi, hareketin tamamını etkilemeden sonralandırabilir.
- **Kurtarılabirlik:** Uzun süren bir hareketin sistem arızası yüzünden iptal edilmesi kabul edilemez bir durumdur. Böyle bir durumda sistem kendini arızadan hemen önceki duruma geri getirebilmeli ve veri kaybına yol açmamalıdır.
- **Başarımlı:** Uzun süreli hareketler içeren sistemler için başarımlı saniyede gerçekleşen hareket sayısı ile ölçülemez. Bu tür sistemler için başarımlı bilgisayar başında çalışan operatör'ün bekletilme zamanı ile ölçümlenir.

5. Gerçek Zamanlı Hareket Sistemleri

Gelişmiş Hareket İşleme

Son zamanı verilmiş ve bir hareketin bu zamandan önce tamamlanacağını garanti edebilen hareket sistemlerine "Gerçek Zamanlı Hareket Sistemleri" denir. Bu tür sistemlere örnek olarak trafik yönetim sistemi, hava alanı iniş-kalkış yönetim sistemi verilebilir. Bu tür sistemlerde önemli olan ne kadar son zamanın aşılmış olduğu değil, son zamanın ne kadar aşılmış olduğudur. 2 tür son zaman mevcuttur. "Katı zaman sınırı" ve "esnek zaman sınırı". "Katı zaman sınırları"nda zaman sınırı kesinlikle aşılmamalıdır. Aksi takdirde bilginin/hareket sonucunun değeri kalmayacaktır. "Esnek Zaman Sınırı"nda ise zaman sınırının aşılması durumunda bilgi hemen değersizleşmeyecek fakat her an daha fazla değer kaybedecektir.

Gerçek zamanlı sistemlerin riskleri normal sistemlere oranla daha farklıdır. Eğer içerik kontrol edici T_i görevinin beklemesine karar verirse T_i 'nin son zamanını kaçırma riski meydana gelir. Bu gibi durumlarda T_i 'nin çalışmasını engelleyen kilidin kaldırılması ve T_i 'nin çalışmasını sağlamak bir çözüm olabileceği gibi bu çözüm kilidi kaldırılan görevin son zamanını kaçırmasına da neden olabilir.

Gerçek zamanlı hareket sistemlerinin de tüm problemi bu noktada meydana gelmektedir. Hangi görevin öncelikli olduğu hangisinin beklerse zaman sınırını kestirmek güçtür.