

1. Dağıtık Sistem Nedir?

Dağıtık sistemler, birden fazla bağımsız bilgisayarın (düğümün) birlikte çalışarak tek bir sistem gibi hareket ettiği bilgisayar sistemleridir. Bu sistemler, fiziksel olarak farklı konumlarda bulunabilir, ancak ortak bir amaca hizmet etmek için bir ağ üzerinden haberleşirler. Özetle: Tek bir büyük bilgisayar yerine, birden fazla makinenin (sunucu, istemci, düğüm) birlikte çalıştığı sistemlerdir. Ağ üzerinden birbirleriyle haberleşerek görev paylaşımı yaparlar. Kullanıcılara merkezi olmayan, hızlı, ölçeklenebilir ve güvenilir bir hizmet sunarlar.

1.1. Dağıtık Sistemlerin Amacı

Dağıtık sistemlerin temel amacı, birden fazla bilgisayarı veya düğümü bir araya getirerek daha güçlü, ölçeklenebilir, güvenilir ve hızlı sistemler oluşturmaktır. Bu sistemler, yüksek performans gerektiren hesaplamalar, hata toleransı, daha hızlı yanıt süreleri ve büyük veri işleme gibi gereksinimlere çözüm sunar. Sistem yükü arttığında, yeni düğümler eklenerek performans artırılabilir. Kullanıcı sayısı, veri miktarı veya iş yükü arttığında, mevcut altyapıyı büyüterek sistemin verimli çalışmasını sağlamak mümkündür. Dağıtık Sistemlere örnek verecek olursak;

Arama Motorları: Arama sorgularını binlerce sunucuya bölerek işler.

Google Cloud, AWS, Microsoft Azure: Bulut tabanlı dağıtık sistemler.

Bitcoin Blockchain: Merkezi olmayan, eşler arası (P2P) bir sistemdir.

Netflix, Youtube: Video içeriklerini küresel sunucu ağına dağıtarak kullanıcılara yakın sunucudan erişim sağlar.

Facebook Messenger: Farklı veri merkezlerinde çalışan mesaj sunucuları ile küresel erişim sunar.

Apache Spark, Hadoop: Büyük veri işleme sistemleri.

2. Dağıtık Sistemlerin Temel Özellikleri

Bir sistemin dağıtık sistem olup olmadığını anlamak için aşağıdaki özellikleri taşıyıp taşımadığına bakılır:

Özellik	Açıklama
Şeffaflık (Transparency)	Kullanıcı, sistemin dağıtık olduğunu fark etmez.
Ölçeklenebilirlik (Scalability)	Kullanıcı veya işlem sayısı arttıkça sistem genişleyebilir.
Hata Toleransı (Fault Tolerance)	Bir düğüm bozulursa sistem çökmemelidir.
Özerklik (Autonomy)	Düğümmler birbirlerinden bağımsız çalışabilir.
Eşzamanlılık (Concurrency)	Aynı anda birçok işlem gerçekleştirilebilir.
Konum Bağımsızlığı (Location Independence)	Kullanıcılar, işlemin nerede gerçekleştiğini bilmez.

2.1. Şeffaflık Türleri

Dağıtık sistemlerde şeffaflık (transparency), kullanıcıların ve uygulamaların sistemin dağıtık yapısını fark etmeden çalışabilmesini ifade eder. Kullanıcı, bir işlemi yerel bir bilgisayarda yapıyormuş gibi hisseder, ancak aslında arka planda birçok düğüm (node) ve sistem birlikte çalışmaktadır. Dağıtık sistemlerin kullanımı kolay olmalıdır. Uygulamalar ve kullanıcılar için altyapının karmaşıklığı gizlenmelidir. Fiziksel konum, veri replikasyonu veya hata toleransı gibi detaylar kullanıcıdan saklanmalıdır. Dağıtık sistemlerde 8 temel şeffaflık türü vardır. Bunlar, sistemin farklı yönlerini kullanıcıdan ve uygulamalardan saklamayı amaçlar.

Şeffaflık Türü	Açıklama
Erişim Şeffaflığı (Access Transparency)	Kullanıcı veya uygulama, kaynağın türünü veya erişim yöntemini bilmeden kullanabilmelidir.
Konum Şeffaflığı (Location Transparency)	Kaynağın fiziksel olarak nerede bulunduğu önemsiz olmalıdır. Kullanıcı bunu bilmek zorunda kalmaz.
Taşıyabilirlik Şeffaflığı (Migration Transparency)	Kaynaklar veya işlemler bir düğümden diğerine taşındığında kullanıcı bunu fark etmez.
Tekrar Başlama Şeffaflığı (Failure Transparency)	Sistem, hata durumlarında kendini otomatik olarak onarır veya yeniler ve kullanıcı bunu hissetmez.
Dağıtım Şeffaflığı (Replication Transparency)	Bir kaynak birden fazla yerde replikasyon ile saklansa bile, kullanıcı tek bir kopyaymış gibi kullanır.
Paralellik Şeffaflığı (Concurrency Transparency)	Aynı kaynağa birden fazla kullanıcı eşzamanlı erişebilir ve tutarsızlık oluşmaz.
Performans Şeffaflığı (Performance Transparency)	Sistem yükü arttığında, performans otomatik ölçeklenebilir ve kullanıcı bunu hissetmez.
Evrim Şeffaflığı (Scaling Transparency)	Sistem yeni düğümler eklenerek genişletildiğinde kullanıcı veya uygulamalar buna müdahale etmek zorunda kalmaz.

3. Merkezi, Dağıtık ve Paralel Sistemler Arasındaki Farklar

Merkezi sistem: Tüm işlem gücünün ve veri yönetiminin tek bir bilgisayar veya sunucuda toplandığı bir sistem modelidir. Tüm kullanıcılar ve uygulamalar tek bir merkezi sunucuya bağlıdır. Veri işleme, depolama ve hesaplama merkezi bir noktada yapılır. Genellikle küçük ölçekli uygulamalarda veya basit veri yönetim sistemlerinde kullanılır. Örneğin; bir şirketin muhasebe yazılımı tek bir bilgisayarda veya sunucuda çalışıyorsa, bu merkezi bir sistemdir. Ya da MySQL veya PostgreSQL gibi geleneksel veritabanları merkezi sistem modeliyle çalışabilir.

Paralel sistemler: birden fazla işlem biriminin (CPU, çekirdek) aynı anda çalıştığı sistemlerdir. Tüm işlem birimleri ortak bir bellek kullanabilir veya kendi belleklerine sahip olabilir. Veri ve işlemler, eşzamanlı olarak birden fazla işlemci tarafından işlenebilir. Paralel sistemler genellikle süper bilgisayarlar veya yüksek performanslı hesaplamalarda kullanılır.

Paralel sistemler, Flynn'in Taksonomisine göre 4 ana gruba ayrılır:

- **SISD (Single Instruction Single Data):** Tek bir işlemci, tek bir veri setini işler (geleneksel bilgisayarlar).
- **SIMD (Single Instruction Multiple Data):** Aynı anda birçok veri seti işlenir (Grafik İşlemciler - GPU'lar).
- **MISD (Multiple Instruction Single Data):** Farklı işlemciler aynı veri setine farklı işlemler uygular (Nadir kullanılır).
- **MIMD (Multiple Instruction Multiple Data):** Bağımsız işlemciler, farklı verileri işler (Süper bilgisayarlar).

Örnek verecek olursak Grafik İşlemciler (GPU) aynı anda birçok pikseli işleyerek görüntüleri oluşturur. Modern CPU'lar birden fazla çekirdeği kullanarak işlemleri paralel çalıştırır. Süper Bilgisayarlar: Büyük ölçekli bilimsel hesaplamalar için paralel işlem gücü kullanır. Merkezi, paralel ve dağıtık sistemler arasındaki farklar aşağıdaki tabloda görülebilir:

Özellik	Merkezi Sistem	Paralel Sistem	Dağıtık Sistem
Mimari	Tek bir bilgisayar	Aynı bellek paylaşılır	Farklı makinelerden oluşur
Bağımsızlık	Bağımsız değil	Bağımsız değil	Bağımsız düğümler var
İletişim	Gereksiz	Paylaşılan bellek	Ağ üzerinden mesaj geçişi
Ölçeklenebilirlik	Düşük	Orta	Yüksek
Hata Toleransı	Düşük	Orta	Yüksek
Örnek Kullanım	Ana çerçeve sistemleri	Süper bilgisayarlar	Google, Amazon, Bitcoin, Netflix

Merkezi sistemler, küçük ölçekli uygulamalar ve tek sunucu tabanlı işlemler için uygundur. Paralel sistemler, büyük ölçekli veri işleme, yapay zekâ, süper bilgisayarlar ve HPC (High-Performance Computing) uygulamaları için kullanılır. Dağıtık sistemler, büyük ölçekli web servisleri, bulut bilişim, blockchain, IoT ve büyük veri analitiği gibi sistemlerde kullanılır.

4. Dağıtık Sistemlerin Avantajları

Dağıtık sistemler, günümüz bilişim dünyasında büyük ölçekli işlemler ve hizmetler sunmak için tercih edilen bir mimari yaklaşımdır. Bu sistemler, yüksek erişilebilirlik, ölçeklenebilirlik, hata toleransı ve performans gibi birçok avantaj sağlar. Bu bölümde dağıtık sistemlerin avantajlarını detaylı bir şekilde inceleyeceğiz.

4.1. Ölçeklenebilirlik (Scalability)

Ölçeklenebilirlik, sistemin artan iş yüküne uyum sağlayarak genişleyebilmesi anlamına gelir. Merkezi sistemlerde, donanım kapasitesi sınırlıdır. Ancak dağıtık sistemlerde yeni düğümler eklenerek kapasite artırılabilir. Sistem, yatay ölçeklenme (horizontal scaling) ve dikey ölçeklenme (vertical scaling) ile genişletilebilir:

- **Yatay Ölçeklenme:** Yeni makineler ekleyerek sistem kapasitesini artırmak (örn. yeni sunucular eklemek).
- **Dikey Ölçeklenme:** Var olan makinelerin donanım özelliklerini artırmak (örn. daha fazla RAM eklemek).

Bir gerçek dünya örneği verecek olursak; Google, artan arama sorgularını işleyebilmek için dünyanın dört bir yanındaki veri merkezlerine sürekli yeni sunucular ekleyerek ölçeklenebilirliği artırır. Trafik artışlarında performans kaybı yaşanmaz. Bununla birlikte, daha fazla düğüm, daha fazla senkronizasyon ve yönetim gerektirir.

4.2. Hata Toleransı ve Güvenilirlik (Fault Tolerance & Reliability)

Hata toleransı, sistem bileşenlerinden biri arızalansa bile sistemin çalışmaya devam edebilmesini sağlar. Dağıtık sistemlerde aynı verinin birden fazla kopyası farklı düğümlerde saklanarak veri kaybı engellenir. Bir düğüm arızalandığında, başka bir düğüm devreye girerek sistemin çalışmaya devam etmesini sağlar. Load balancer (yük dengeleme) sayesinde, arızalı düğümlere istek yönlendirilmez. Örneğin; Netflix, içeriklerini farklı veri merkezlerinde saklayarak herhangi bir sunucuda sorun yaşandığında, kullanıcıları başka bir sunucuya yönlendirir. Hata toleransı sayesinde Kesintisiz hizmet sunulur ve veri kaybı önlenir. Bununla birlikte replikasyon ve hata yönetimi için ek donanım ve yazılım gerektirir.

4.3. Performans Artışı (Increased Performance)

Dağıtık sistemler, birden fazla düğümün paralel çalışmasını sağlayarak daha hızlı işlem yapar. İş yükü birden fazla düğüme bölündüğünde, her düğüm belirli bir görevi yerine getirir. Merkezi sistemlerde tek bir sunucu yoğun trafik altında yavaşlarken, dağıtık sistemler yükü çoklu düğümler arasında bölerek işleyebilir. Özellikle paralel işlem gerektiren büyük veri analizleri, yapay zeka ve bilimsel hesaplamalarda büyük avantaj sağlar. Örneğin; Google Haritalar, bir güzergah hesaplamak için dünyanın dört bir yanındaki sunucularını kullanarak saniyeler içinde sonuç döndürebilir. Daha hızlı yanıt süreleri ve yüksek işlem gücü getirdiği açıkça görülebilir. Ancak, ağ gecikmesi oluşabilir ve veri senkronizasyonu zor olabilir.

4.4. Kaynak Paylaşımı (Resource Sharing)

Dağıtık sistemler, işlem gücü, veri depolama ve bellek gibi kaynakları paylaşabilir. Kaynak paylaşımı sayesinde verimli kullanım sağlanır ve maliyetler düşer. Aynı anda birden fazla kullanıcı sisteme bağlanabilir ve her kullanıcı farklı bir düğümden hizmet alabilir. Bulut bilişim altyapıları (AWS, Google Cloud, Microsoft Azure), kaynak paylaşımıyla büyük ölçekli hizmetler sunar. Örneğin; Google Drive ve Dropbox gibi hizmetler, farklı sunucularda depolanan verileri kullanıcılara şeffaf bir şekilde sunar. Kaynak paylaşımı daha düşük maliyet ve daha verimli kullanım hizmeti sunar, ancak verilerin paylaşımı sırasında güvenlik riskleri oluşabilir.

4.5. Düşük Gecikme (Low Latency)

Dağıtık sistemler, verileri kullanıcılara en yakın düğümden sunarak gecikmeyi azaltır. İçerik Dağıtım Ağları (CDN - Content Delivery Networks) sayesinde, kullanıcılar verilere en yakın sunucudan erişebilir. Bu sayede, uzun mesafelerden gelen verilerin yol açtığı yüksek gecikme süreleri (high latency) düşürülmüş olur. YouTube ve Netflix, içeriklerini farklı ülkelerdeki sunuculara dağıtarak videoların hızlı yüklenmesini sağlar. Bu özellik sayesinde kullanıcı deneyimi iyileşir, hız artar. Ancak, farklı veri merkezleri arasında veri tutarlılığı sağlamak zor olabilir.

4.6. Güvenlik ve Veri Gizliliği (Security & Data Privacy)

Dağıtık sistemler, merkezi olmayan güvenlik yöntemleri ile saldırılara karşı daha dayanıklı olabilir. Merkezi sistemlerde tek bir sunucu saldırıya uğradığında tüm sistem zarar görebilir. Dağıtık sistemlerde veriler farklı düğümlerde saklandığı için, bir noktada yapılan saldırı tüm sistemi etkilemez. Şifreleme (Encryption), güvenli erişim (TLS, VPN) ve kimlik doğrulama (OAuth, JWT) gibi teknolojiler ile güvenlik artırılabilir. Bitcoin ve Ethereum gibi blockchain tabanlı dağıtık sistemler, merkezi otoriteler olmadan güvenliği sağlar. Dağıtık sistemlerde saldırılara karşı daha dirençli bir yapı oluşturulabilir ve veri gizliliği sağlanabilir. Ancak, merkezi bir yönetim olmaması, güvenlik açıklarına neden olabilir.

5. Dağıtık Sistemlerin Zorlukları

Dağıtık sistemler ölçeklenebilirlik, hata toleransı, performans artışı gibi birçok avantaj sunmasına rağmen, bazı karmaşık zorlukları da beraberinde getirir. Bu zorluklar sistem tasarımı, senkronizasyon, veri tutarlılığı, ağ iletişimi ve güvenlik gibi alanlarda ortaya çıkar. Bu bölümde, dağıtık sistemlerin karşılaştığı temel zorlukları detaylı bir şekilde inceleyeceğiz.

5.1. Ağ Gecikmesi ve İletişim Problemleri (Network Latency & Communication Issues)

Dağıtık sistemler, ağ bağlantıları aracılığıyla düğümler arasında iletişim kurar. Ancak bu, ağ gecikmesi (latency) ve paket kaybı gibi sorunlara yol açabilir. Farklı coğrafi bölgelerdeki düğümler arasında veri iletimi gecikmelere neden olabilir. Ağ tıkanıklıkları, sistemin yanıt sürelerini artırabilir. Paket kayıpları ve ağ bağlantı kopmaları, veri bütünlüğünü bozabilir. Örneğin; bir kullanıcı Netflix üzerinden video izlerken, en yakın sunucudan veri almazsa, gecikme artar ve buffering (yükleme süresi) yaşanır. Amazon gibi e-ticaret siteleri, farklı bölgelerdeki veri merkezlerinden en hızlı yanıt verecek olanı belirlemek zorundadır. Bu zorluğun üstesinden gelmek için CDN (Content Delivery Network) teknolojisi kullanarak veriler en yakın sunucudan sağlanır.

5.2. Veri Tutarlılığı (Data Consistency)

Dağıtık sistemlerde verilerin birden fazla düğümde saklanması, veri tutarlılığı (data inconsistency) problemlerine yol açabilir. Aynı veri farklı düğümlerde güncellendiğinde, hangi sürümün doğru olduğu belirlenmelidir. Tutarlılık, erişilebilirlik ve ağ bölünmesi arasında denge kurulması gerekir (CAP Teoremi). Eventual Consistency (Sonradan Tutarlılık) modelinde, veriler zamanla senkronize olur, ancak anlık olarak tutarlılık yaşanabilir. Amazon DynamoDB gibi dağıtık veritabanları, veri güncellemelerini bir süre sonra diğer düğümlere yansıtır. WhatsApp mesajları farklı cihazlarda senkronize edilirken bazen gecikmeler yaşanabilir. Bu tür problemleri aşmak için Tutarlılık Protokolleri kullanılır (Strong Consistency, Eventual Consistency). Ayrıca Quorum bazlı güncelleme yöntemleri de kullanılabilir (örneğin, Paxos veya Raft algoritmaları).

5.3. Senkronizasyon (Synchronization)

Aynı anda çalışan birden fazla işlem arasında veri çakışmaları olabilir. Farklı düğümlerde çalışan süreçler, ortak bir veri kümesine eriştiğinde çakışmalar yaşanabilir. Dağıtık kilitleme (distributed locking) mekanizmaları karmaşıktır. Veri tutarlılığına neden olmamak için işlemlerin senkronize edilmesi gerekir. Örneğin; banka sistemlerinde aynı hesaptan iki farklı ATM'den para çekme işlemi çakışmalara neden olabilir. Ya da e-ticaret sitelerinde aynı ürünü iki farklı kullanıcının satın alması, stok yönetimi açısından sorun yaratabilir. Bu zorluğun üstesinden gelmek için genelde Zookeeper, Etcid gibi dağıtık koordinasyon araçları kullanılır. Ayrıca Lamport Zaman Damgaları (Logical Clock) gibi teknikler de kullanarak işlemler sıralanabilir.

5.4. Güvenlik Problemleri (Security Challenges)

Merkezi sistemlerde güvenliği sağlamak daha kolaydır, ancak dağıtık sistemlerde güvenlik daha karmaşıktır. Ağ üzerinden iletişim kurulduğu için veriler saldırılara açık hale gelir. Kimlik doğrulama ve yetkilendirme yönetimi zorlaşır. DDoS saldırıları (Dağıtık Hizmet Reddi Saldırıları) sistemi çökertme riski taşır. Bitcoin gibi blockchain sistemlerinde, madencilik saldırılarında (51% attack) büyük risk vardır. AWS ve Google Cloud gibi bulut sağlayıcıları, sürekli siber saldırılara karşı önlem almak zorundadır. Bu problemi aşmak için şu işlemler yapılır; (i) TLS/SSL şifreleme kullanmak, (ii) JWT (JSON Web Token) veya OAuth gibi kimlik doğrulama sistemleri uygulamak, (iii) DDoS saldırılarına karşı bulut tabanlı koruma (Cloudflare, AWS Shield) kullanmak.

5.5. Hata Yönetimi ve Hata Toleransı (Fault Handling & Fault Tolerance)

Dağıtık sistemlerde bir düğüm çöktüğünde, sistemin çalışmaya devam etmesi gerekir. Tüm düğümlerin aynı anda hatasız çalışması beklenemez. Bir düğüm çökerse, diğer düğümlerin bunu fark edip devreye girmesi gerekir. Hata toleransı sağlamak için sistemin otomatik olarak kendini onarması gerekir. Amazon AWS'de bir veri merkezi çökse bile, başka bir veri merkezi otomatik olarak devreye girer. Netflix, bir sunucu çökse bile içeriği diğer sunuculardan sunmaya devam eder. Replikasyon (Data Replication) kullanarak veriyi yedeklemek, Leader Election algoritmaları (Raft, Paxos) ile düğümler arasında yönetimi devretmek, Self-healing sistemler (Kubernetes gibi) kullanarak hataları otomatik olarak onarmak vb. işlemlere bu sorun çözülebilir.

5.6. Merkezi Yönetim Eksikliği (Lack of Centralized Control)

Dağıtık sistemlerde merkezi bir kontrol mekanizması yoktur, bu da yönetimi zorlaştırır. Sistemin farklı düğümleri bağımsız çalıştığı için denetim zorlaşır. Hata ayıklama (debugging) ve sistemin izlenmesi daha karmaşıktır. Merkezi olmayan sistemlerde, karar mekanizmaları dağıtılmış olduğu için karar alma süreci uzayabilir. Google veya Facebook gibi dev şirketler, farklı kıtalardaki veri merkezlerini uyum içinde çalıştırmak için otomasyon sistemleri kullanır. Dağıtık kontrol sistemleri (Raft, Zookeeper) bu otomasyon sistemlerine örnektir. Bir otomasyon sistemi kullanılmıyacaksa monitoring araçları (Prometheus, Grafana) ile sistemin durumunu sürekli takip edilebilir.

6. CAP Teoremi

CAP Teoremi, dağıtık sistemlerde aynı anda üç temel özelliğin (Consistency-Tutarlılık, Availability-Erişilebilirlik, Partition Tolerance - Bölünme Toleransı) tam olarak sağlanamayacağını ifade eden bir teoremdir. Teorem, Eric Brewer tarafından 2000 yılında ortaya atılmış ve 2002'de Seth Gilbert ve Nancy Lynch tarafından matematiksel olarak kanıtlanmıştır. Özetle: Bir dağıtık sistem, aynı anda üç özelliğin tamamına sahip olamaz; en fazla ikisini sağlayabilir. Bu üç özellik aşağıda yer almaktadır:

Özellik	Açıklama
Tutarlılık (Consistency)	Tüm düğümler her zaman en güncel veriyi görmelidir.
Erişilebilirlik (Availability)	Her istek bir yanıt almalıdır (hata mesajı yerine).
Bölünme Toleransı (Partition Tolerance)	Ağda bölünmeler (network failure) olduğunda bile sistem çalışmaya devam etmelidir.

CAP Teoremine göre, bir dağıtık sistem aynı anda en fazla iki özelliği tam olarak sağlayabilir:

Seçenek	Hangi İki Özellik Sağlanıyor?	Hangi Özellikten Ödün Veriliyor?	Örnek Sistemler
CP (Consistency + Partition Tolerance)	Veriler her zaman tutarlıdır, ağ bölünse bile güncellik korunur.	Sistem zaman zaman erişilemez olabilir.	Zookeeper, HBase, BigTable
AP (Availability + Partition Tolerance)	Sistem her zaman çalışır ve her istek bir yanıt alır.	Veri bazen eski olabilir (eventual consistency).	DynamoDB, Cassandra, CouchDB
CA (Consistency + Availability) [Uygulanamaz]	Tüm düğümler güncel veri tutar ve her istek bir yanıt alır.	Ağ bölünmesi yaşanamaz. (Gerçek dünyada mümkün değil!)	Teorik olarak var ama gerçek dünyada uygulanamaz.

6.1. CAP Teoremine Göre Sistem Türleri

CAP Teoremine göre, dağıtık sistemlerin nasıl çalıştığını anlamak için iki temel yaklaşım vardır:

CP (Consistency + Partition Tolerance) Sistemleri: Tutarlılığı ön planda tutar, ancak sistem bazen erişilemez olabilir. Ağ bölünmesi yaşandığında, sistem veri bütünlüğünü sağlamak için belirli işlemleri durdurabilir. Bankacılık sistemleri, finans uygulamaları gibi hassas verilerin olduğu yerlerde tercih edilir. Örneğin Zookeeper uygulaması dağıtık koordinasyon sistemidir, lider seçme mekanizması sayesinde tutarlılığı garanti eder. HBase & BigTable gibi büyük veri mimarileri de dağıtık veritabanlarında tutarlılığı korumak için kullanılır.

AP (Availability + Partition Tolerance) Sistemleri: Erişilebilirliği ön planda tutar, ancak veriler bazen güncel olmayabilir (eventual consistency). Ağ bölünmesi yaşandığında bile sistem çalışmaya devam eder, ancak farklı düğümler geçici olarak farklı veriler gösterebilir. Sosyal medya, e-ticaret, IoT gibi hızlı yanıt gerektiren sistemlerde kullanılır. Amazon DynamoDB, bu yaklaşımı kullanarak tüm işlemlerin çalışmaya devam etmesini sağlar, ancak bazen farklı düğümlerde eski veriler görülebilir. Apache Cassandra uygulaması da bu yaklaşım ile büyük ölçekli veri sistemleri için optimize edilmiştir ve erişilebilirlikten ödün vermez.