

Modern Dağıtık Dosya Sistemlerinin Yapısal Karşılaştırılması

Bahadır KARASULU
Serdar KORUKOĞLU

Ege Üniversitesi
AB2008 ~ 30 Ocak - 01 Şubat 2008

Motivasyon

- Giriş
- Dağıtık dosya sistemlerinin genel yapısı
- Yaygın kullanılan dağıtık dosya sistemleri
 - *Sun Ağ Dosya Sistemi (NFS)*
 - *CMU CODA (Andrew Dosya Sistemi'nin takipçisi)*
 - *Plan 9 ve XFS*
- Güncel bazı dağıtık dosya sistemleri
 - *OpenAFS*
 - *GoogleFS*
 - *Microsoft Dağıtık Dosya Sistemi (MS-DFS)*
- Yedi adet Dağıtık Dosya Sisteminin karşılaştırılması
- Sonuçlar

Giriş

- İlk nesil Dağıtık Dosya Sistemleri, 1970'lerin ortalarında geliştirilmeye başlanmıştır.
- İlk yapılar daha çok FTP-benzeri, ağ ile bağlantılandırılmış depolama sistemleri temellidir.
 - İlk örnekler arasında Svobodava [1] 'nın ve Swinehart vd. [2] çalışmalarında bahsedilen; WoodstuckFS, XDFS, SWALLOW, LOCUS, ACORN ve CMU (Carnegie Mellon Üniversitesi)'ya ait VICE sayılabilir.
- Günümüze yaklaştıkça,
 - Sun Microsystems'e ait Ağ Dosya Sistemi (NFS) ve CMU'ya ait Andrew Dosya Sistemi (AFS) gibi yaygın kullanılan sistemler ortaya çıkmıştır.

DAĞITIK DOSYA SİSTEMLERİNİN GENEL YAPISI

(1/2)

- Dağıtık dosya sistemleri intranet (veya Internet) üzerinden bilgi paylaşımının dosya biçiminde yapılmasını sağlamaktadırlar.
- Bir dağıtık dosya sistemi programların uzaktaki dosyalara yerel dosyalar gibi erişebilmesi ve onları depolayabilmelerine olanak tanır, bu sayede intranet (veya Internet) üzerindeki herhangi bir bilgisayardan kullanıcılar bu dosyalara erişebilir.
- Disk organizasyonlarının avantajları ve anahtarlamalı yerel ağların yüksek bantgenişliği bağlanabilirlikleri sayesinde yüksek performans ve yüksek dereceden ölçeklenebilir sistemler oluşturulabilmektedir.
- Not:
 - **UNIX semantikleri:** Bir dosya üzerindeki her bir işlem tüm süreçler tarafından anında görülebilir durumdadır.
 - **Oturum semantikleri:** Dosya kapatılana kadar hiçbir değişiklik diğer süreçler tarafından görülebilir değildir.

DAĞITIK DOSYA SİSTEMLERİNİN GENEL YAPISI

(2/2)

- Dağıtık dosya sistemleri hataya karşı dayanıklılıklarına ve paralel çalışmaya uygunluklarına göre değerlendirilebilirler.
- Hataya dayanıklılık sınıfı örneği olarak:
 - CMU'ya ait CODA ve Microsoft Corporation'a ait MS-DFS (Dağıtık Dosya Sistemi)
- Hem paralel çalışmaya uygunluk (yüksek başarımli hesaplama) hem de hataya dayanıklılık sınıfı örneği olarak:
 - Google firmasının geliştirdiği Google Dosya Sistemi (GoogleFS)

NFS (Ağ Dosya Sistemi)

- Sun Microsystems tarafından ilk olarak 1985'te geliştirildi.
- En popüler, açık ve yaygın kullanımda olan dağıtık dosya sistemi.
- NFS protokolü bir standart haline getirilmiştir.

AFS (Andrew Dosya Sistemi)

- CMU tarafından Andrew dağıtık hesaplama ortamı'nın parçası olarak geliştirildi (1986)
- Kampus çapında dosya sistemi oluşturmayı hedefleyen bir araştırma projesiydi.
- Daha sonra Linux üzerinde LinuxAFS ve en son olarak OpenAFS isimli (Windows-MAC OSX e kadar) gerçekleştirmeleri bulunmaktadır.
- Temel olarak DCE/DFS sistemini alır. Bu sistem, Açık Yazılım Vakfı (OSF, www.opengroup.org) tarafından geliştirilen bir DEC (Dağıtık Hesaplama Ortamı) dosya sistemidir.
- CODA, AFS versiyon 2'nin takipçisidir.

CODA

Coda AFS v2'yi temel alan bir dağıtık dosya sistemidir. Ağ dosya sistemleri tarafından istenilen özellikler yanında belli başlı onu önplana çıkaran özelliklere de sahiptir. Bunlar:

1. Mobil hesaplama için bağlantı kopartılmışken bile çalışabilmesi,
2. Liberal bir lisans altında bedava olarak elde edilebilmesi,
3. İstemci-tarafli kalıcı önbellekleme sayesinde yüksek performans sunması,
4. Sunucu replikasyonu,
5. Yetkilendirme, şifreleme ve erişim kontrolü için bir güvenlik modeli önermesi,
6. Sunucu ağı'ndaki kısmi ağ hataları boyunca çalışmasını sürdürür ve bantgeniřliđi uyarlanabilirliđini sađlaması,
7. İyi bir ölçeklenebilirliđe sahip olması,
8. Ağ hataları olsa bile, iyi tanımlanmış paylaşım semantiklerine sahip olduđu için sorun oluşmaması,

olarak sayılabilir.

Plan 9

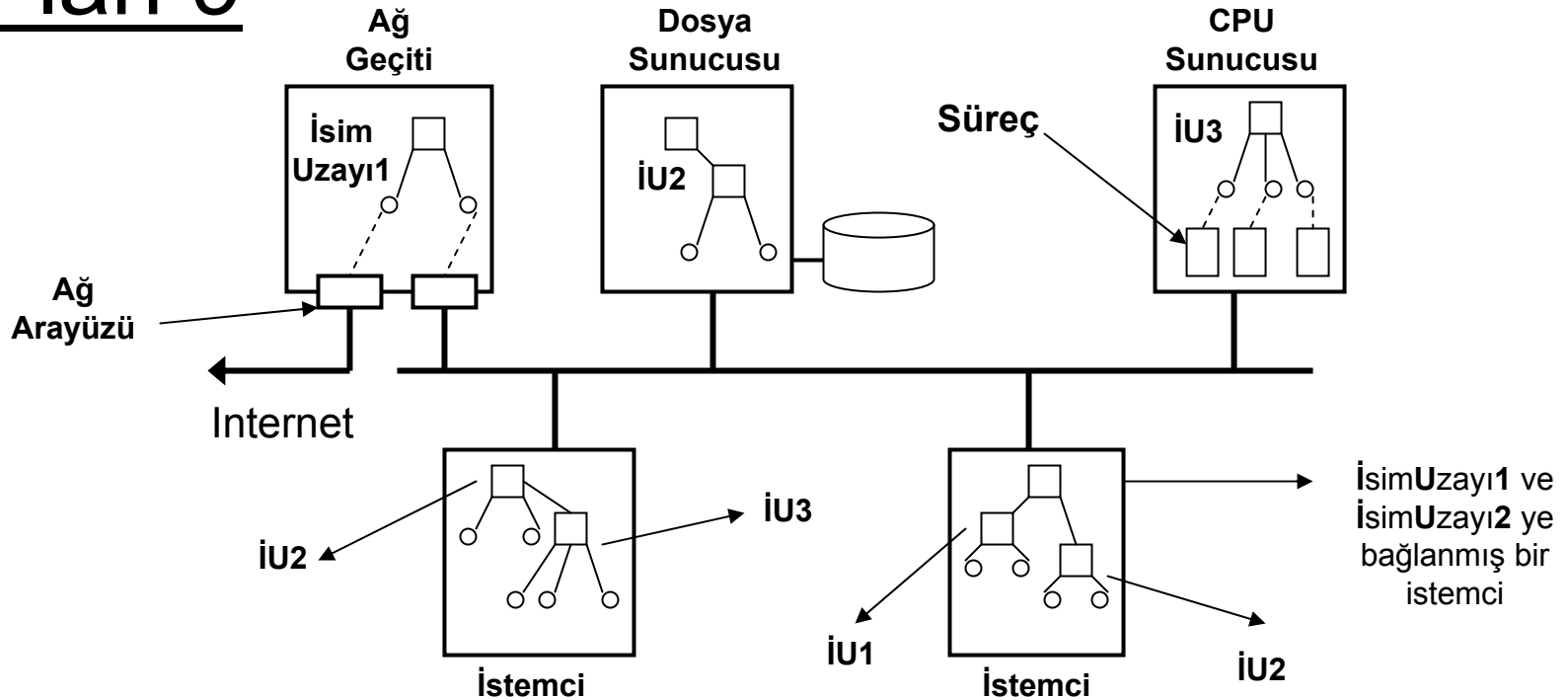
Bell Laboratuvarları tarafından ağ işletim sistemleri'nin 1980'lerde kazandığı popülerliğe tepki olarak geliştirilmiştir.

- Plan 9 tasarım amaçları:
 - **Kaynakların dosyalar olarak gösterilmesi:**
Tüm kaynaklar hiyerarşik bir dosya sisteminde dosyalar olarak gösterilir.
 - **İsim uzayları:**
Uygulama bakış açısıyla ağ, tek bir uyumlu hiyerarşik dosya sistemi olarak gözükür, fakat bu sistem (yerelde veya uzakta bulunan) fiziksel olarak ayrılaştırılmış kaynakları temsil edebilir,
 - **Standart iletişim protokolü:**
9P adı verilen standart bir protokol kullanılarak, hem yerel hem de uzaktaki kaynaklara ulaşılabilir.

Yaygın kullanılan dağıtık dosya sistemleri

(4/5)

Plan 9



Plan 9 sisteminin genel organizasyon şeması.

XFS (Sunucusuz Dosya Sistemi)

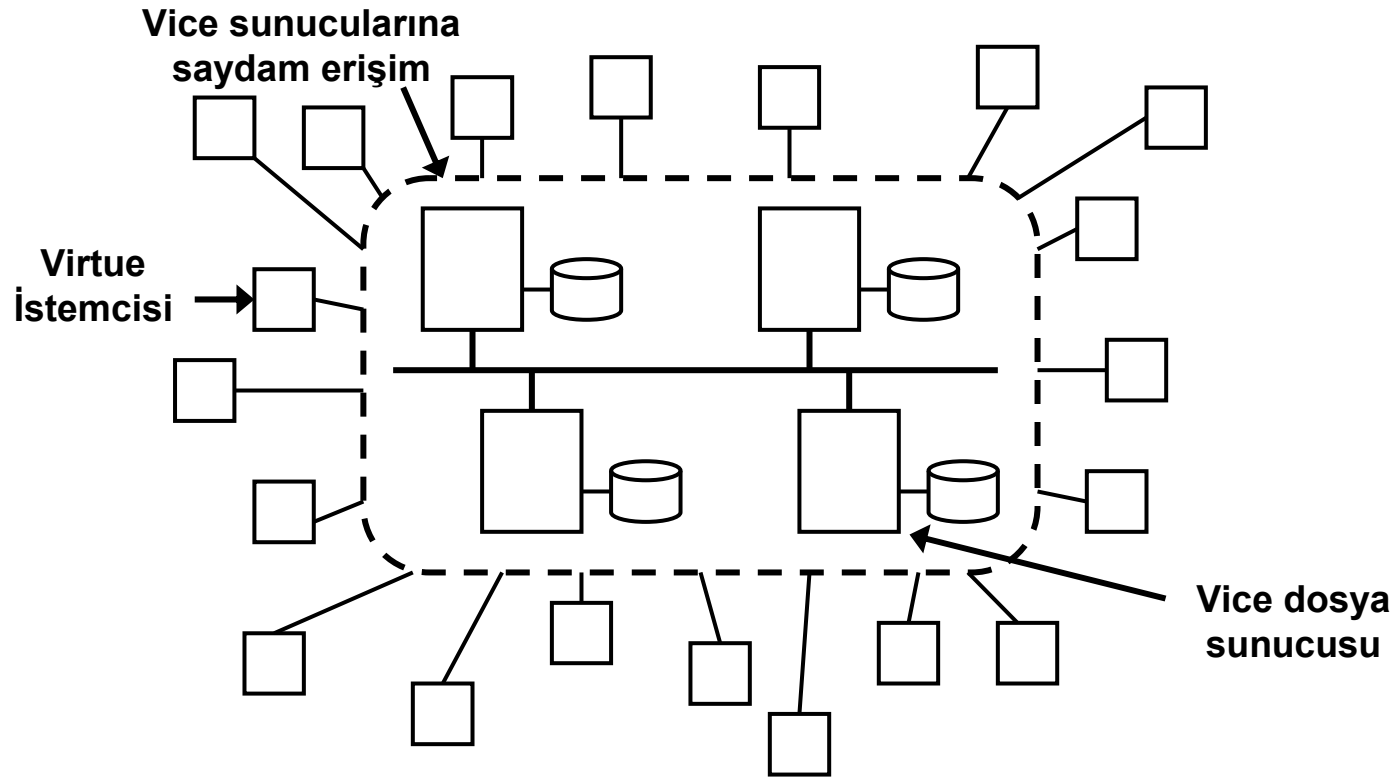
- Berkeley Üniversitesi tarafından geliştirilmiştir.
- Sunucusuz dosya sistemleri, dosya sistemi sunucusu cevaplarını büyük sayıda ortaklaşa çalışan makineler arasında dağıtır. Bu yaklaşım günümüz dosya sistemlerinin tasarımlarının doğası gereği performans iyileştirmesi, ölçeklenebilirlik ve elde edilebilirlik sağlanmasında merkezi bir sunucunun yaratacağı darboğazın yok edilmesini sağlar.
- Sunucusuz sistemler maliyet etkin sistemlerdir. Ölçeklenebilir mimarisi özelleştirilmiş sunucu donanımı ve dosya sisteminin yönetsel gerekliliklere ihtiyacını yok eder.
- XFS mimarisinde, üç farklı tipte süreç bulunur.
 - Bir **depolama sunucusu**, bir dosyanın belirli kısımlarını depolamaktan sorumludur.
 - Bir **üstveri yöneticisi**, bir dosya veri bloğunun gerçekte nerede depolandığının takip edilmesinden sorumludur.
 - Bir **istemci**, XFS'de dosyalar üzerinde çalışmak için kullanıcı isteklerini kabul eden bir süreçtir.
- XFS'te önceleri tüm iletişim RPC mesajlaşmasıyla olurken daha sonra bu yaklaşım terk edilerek (performans kaybı nedeniyle) **aktif mesaj** yaklaşımına geçilmiştir. Buna göre, alıcı taraflı tanımlayıcı, çağrı yapmak için gerekli parametreler ile belirlenir

OpenAFS

- AFS sistemini temel alan açık-kaynak kodlu bir dağıtık dosya sistemi gerçekleştirimidir.
- Şu anda IBM Pittsburgh Laboratuvarları tarafından geliştirilmeye devam edilmektedir. OpenAFS geniş çapta UNIX, Linux, MacOS X ve Microsoft Windows sistemlerini destekler.
- AFS, sunucular üzerindeki yükün azaltılmasını, büyük boyutlu chunk'lar veya bir bütün dosyanın önbelleklenmesi yoluyla sağlamaya çalışır. Bunun için ölçeklenebilir bir ortam yaratır.
- AFS ve OpenAFS dosya sistemi tüm istemciler için tekdüze bir isim uzayı yaratarak onları **"/afs"** dizini altına ekler. Bu sayede kullanıcılar aynı sistem bakış açısıyla farklı işstasyonlarında dolaşabilirler.

Güncel bazı dağıtık dosya sistemleri (2/8)

- CODA sisteminin genel organizasyon şeması, AFS'ye benzer.



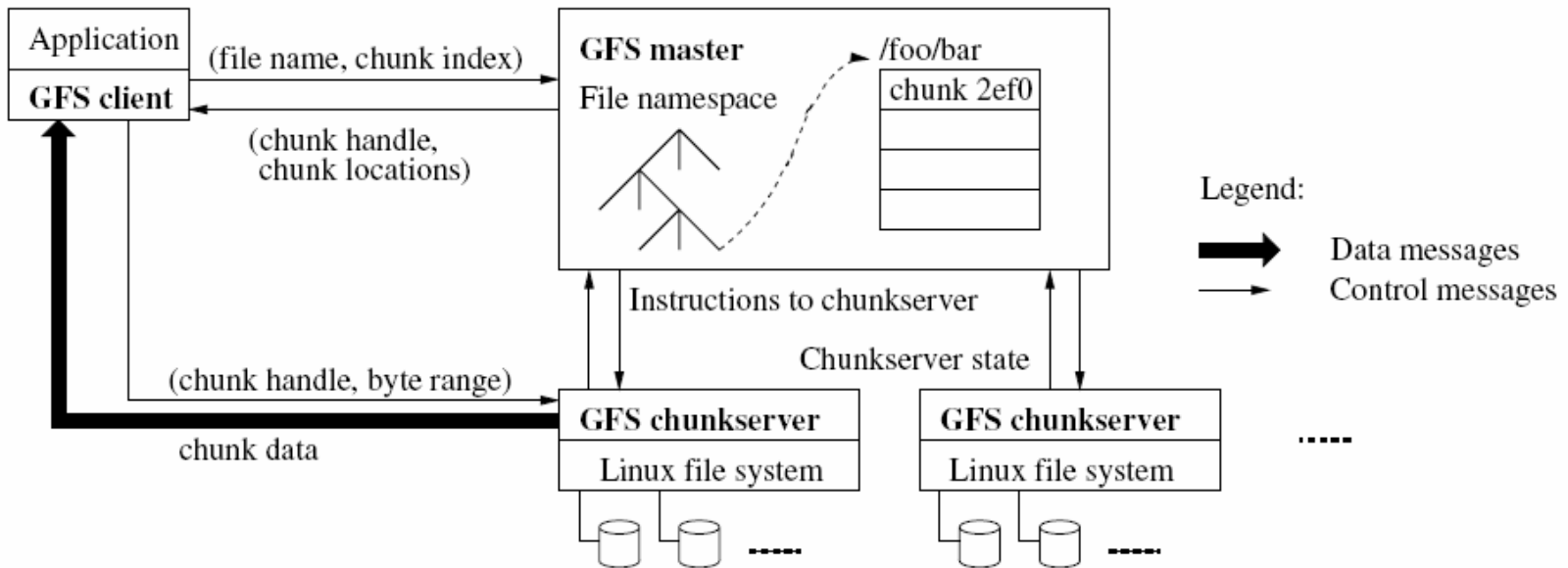
Google Dosya Sistemi (GoogleFS)

- Google firması tarafından yüksek depolama isterlerini en verimli yoldan (büyük miktarda veri işlemek için) elde edebilmek adına tasarlanan ölçeklenebilir dağıtık dosya sistemidir.
- Ücretsiz olarak dağıtılmamaktadır.
- Diğer sistemlere benzer olarak hiyerarşik yapıda düzenlenmiş dizinler ve dosyalarla ilgilenir, fakat bunların haricinde snapshot ve kayıt ekleme işlemlerini de içerir.
 - **Snapshot**, bir dosya veya dizin ağacının en düşük maliyetle bir kopyasının yaratılmasıdır.
 - **Kayıt ekleme** ise, her bir tekil istemcinin eklenmesinin garanti edildiği durumda, aynı dosyaya eşzamanlı olarak veri ekleyen birden çok istemciye izin vermektir.

Google Dosya Sistemi (GoogleFS)

- Bir GoogleFS kümesi, tek bir master ve birden çok chunk sunucu bulundurur ve birden çok istemci tarafından erişilebilir durumdadır.
- Dosyalar sabit uzunluklu (64 Megabyte) chunk'lara bölünmüştür. Her bir chunk global olarak tekil 64 bit chunk handle ile tanımlanmıştır. Bu tanımlayıcı master sunucuda chunk yaratılırken atanır. Güvenlik için replikasyon yapılır.
- Master sunucu tüm dosya sisteminin üstveri'sini tutar.
- Master sunucu periyodik olarak her bir chunk sunucusuyla HeartBeat (Kalp atışı) ismi verilen mesajlarla iletişime geçerek, onların durumlarını kontrol eder.

Google Dosya Sistemi (GoogleFS)



Google Dosya Sistemi mimarisini gösteren şema.

Microsoft Dağıtık Dosya Sistemi (MS-DFS)

- Microsoft firmasına aittir. Ücretsiz olarak dağıtılmamaktadır.
- MS-DFS yerleşim saydamlığını, hatanın varlığı veya birden çok paylaşımın olduğu durumlardaki yoğun yük altında veri elde edilebilirliğinin iyileştirilmesini amaçlar.
- Kullanıcılar DFS'nin root dizininde olmayan bir paylaşımına erişmeye çalıştıklarında kullanıcı gerçekte bir DFS link'ine bakmakta ve DFS sunucusu saydam olarak onu doğru dosya sunucusuna yönlendirmekte ve paylaşım sağlanmaktadır.
- Windows 2000 sadece sunucu başına bir DFS root'u bulundurabilirken, Windows Server 2003 birden çok DFS root'unu aynı sunucuda bulundurabilmektedir.
Bir DFS'nin root'unu bir Samba sunucusu da bulundurabilir.

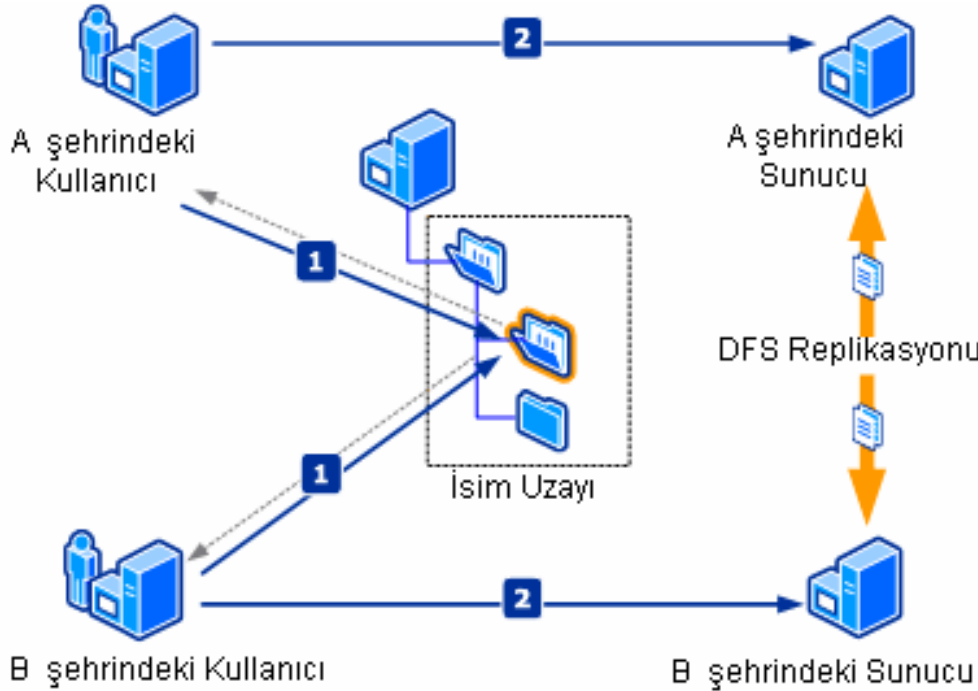
Microsoft Dağıtık Dosya Sistemi (MS-DFS)

- Windows 2000 ve Windows Server 2003'te DFS gerçekleştirimi için iki yol bulunmaktadır:
 - Birinci olarak; **tek başına (standalone)** bulunan DFS root'ların var olduğu durumda sadece bir yerel bilgisayarda var olan bir DFS root'a izin verilir ve bu aktif dizin yapısını kullanmaz. Bir tek başına bulunan DFS sadece yaratıldığı makineden erişilebilirdir, herhangi bir hataya dayanıklılık ve diğer DFS'lere bağlantı da önermemektedir.
 - İkinci olarak; **domain-tabanlı** DFS root'ları bir aktif dizin içerisinde bulunmakta ve domain içindeki diğer domain kontrolcileri ile bilgilerini paylaşmakta, yani onlara bilgi dağıtmaktadır. Bu sayede DFS'ye hataya dayanıklılık özelliği kazandırılmış olur.

MS- DFS nin başlıca özellikleri:

- **DFS Replikasyonu:** MS-DFS, yeni durum-tabanlı multimaster replikasyon motoru ile WAN ortamları için optimize edilmiştir. Bandgenişliği ayarlamaları ve yeni bir byte-seviyesi sıkıştırma algoritması (Uzak Farksal Sıkıştırma (RDC)) kullanılmaktadır.
- **DFS İsimuzayları:** MS-DFS, Farklı sunucularda bulunan paylaşımlı dizinlerin kullanıcılara isimuzayı denilen sanal bir dizin ağacı şeklinde sunulmasını sağlar. Sistem için bu durum Dağıtık Dosya Sistemi olarak algılanır.

Microsoft Dağıtık Dosya Sistemi (MS-DFS)



İzlenen süreç'e bir bakalım:

1-) Bir kullanıcı isim uzayını hedefleyen bir klasöre eriştiğinde, istemcinin bilgisayarı isimuzayı sunucusuna bağlantı kurarak oradan bir referans alır.

2-) Kullanıcı (istemci) bu şekilde aldığı referansla alakalı ilk sunucuya erişir.

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

- NFS ve CODA 'nın mimari ve iletişim, isimlendirme ve senkronizasyon, hataya karşı dayanıklılıklarını detayları ile inceleyelim...

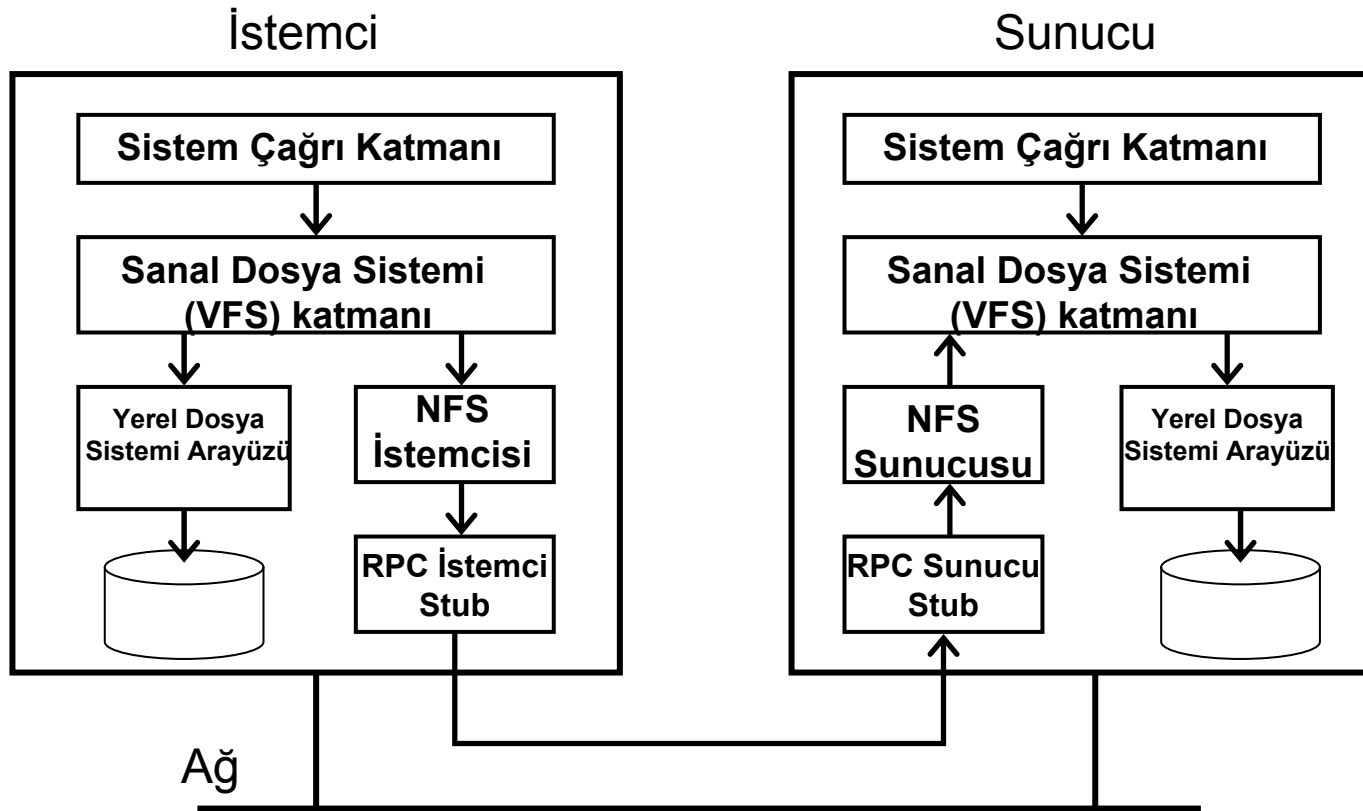
En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

● Sun NFS (Ağ Dosya Sistemi) Mimari ve İletişim:

- NFS'nin altında yatan model, uzak dosya servisi'dir. Bu model ile istemcilere uzaktaki bir sunucu tarafından yönetilen dosya sistemine saydam bir erişim önerilmektedir. İstemciler aslında dosyanın gerçek yerleşiminden habersizlerdir.
- Dosya sistemine erişmek isteyen bir istemcinin ilk önce kendi yerel işletim sistemine sistem çağruları ile erişmesi gerekir, bunu yaparken dağıtık dosya sistemleri için *de facto* standart olan Sanal Dosya Sistemi (VFS) arayüzünü kullanması gerekir.
- NFS 'de bir sunucu ile bir istemci arasındaki tüm iletişim Açık Ağ Hesaplama RPC (ONC RPC) protokolü ile yapılmaktadır

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

- Sun NFS (Ağ Dosya Sistemi) Mimari ve İletişim:



En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

● Sun NFS (Ağ Dosya Sistemi) Mimari ve İletişim:

- NFS için protokol ile istemci-sunucu etkileşiminde v2 ve v3'e kadar durumsuz (stateless) sunucu yaklaşımını önerilirken, NFS v4'te bundan vazgeçilmiştir. Bu durumun tek avantajı basit oluşudur. NFS v4'te durumlu (stateful) sunucu tipine geçilmiştir.
- NFS v4, birkaç RPC'nin tek bir isteğe gruplanabildiği bileşen prosedürlerini desteklemektedir.

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

● Sun NFS (Ağ Dosya Sistemi)

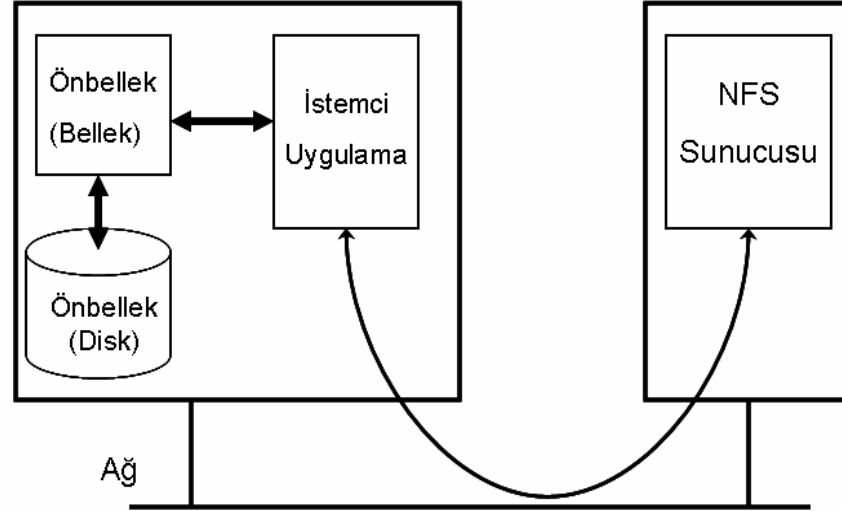
İsimplendirme ve Senkronizasyon:

- Tüm uzak dosya sistemini bağlamak yerine, ilgili parçanın bağlanmasının daha etkin olacağı göz önüne alınmıştır.
- Look-up işleminde iteratif isim çözme işleminin maliyetinin yüksek oluşuna etkin bir çözüm olarak NFS v4'te çapraz bağlanma noktalarının look-up işlemi ile bakılması yoluyla bu problem çözülmüştür.
- NFS için, istemci makinesinde ayrı bir süreç ile çalıştırılan *Otomatik Bağlayıcı* (Automounter) sayesinde uzak bir dosya sistemini anında bağlamak mümkün olmaktadır.
- Dağıtık dosya sistemlerinde tutarlılık senkronizasyonu önemli bir konudur. Dosyalar merkezi bir sunucuda tutulursa tutarlılığı sağlamak görece kolaydır, fakat bu performans problemlerine yol açar.

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

● Sun NFS (Ağ Dosya Sistemi) İsimlendirme ve Senkronizasyon:

- Dosya kapatma sırasında güncel veri'nin sunucuya gönderilmesi temeline dayalı oturum semantik kuralı çoğu dağıtık dosya sisteminde olduğu gibi NFS'de de bulunur. Bu upload/download temelli bir modeldir. Dosya üzerinde en son değişiklik yapan en güncel veriyi sunucuya göndermiş olur.
- Yandaki şekilde istemci uygulama ağ üzerinden NFS sunucusuna erişip veri paylaşımı yapmaktadır.



En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

● Sun NFS (Ağ Dosya Sistemi)

Hataya karşı dayanıklılık:

- NFS sistemi temelde durumsuz (stateless) sunucuyu desteklediği için, her RPC iletişim aşamasında sorunla karşılaşmak olasıdır.
- Herhangi bir sistem çökmesinde bir durum kaybı olmamaktadır, fakat (stateful) kilitleme yöneticileri istemci durumları hakkında bilgi sahibi olmak isterler.
- RPC isteklerinin kaybolması durumunda tekrar tekrar gönderilmesi bir problem oluşturduğu için, sunucu-tarafında çift kez gönderilmiş istek önbelleği kullanımı ve her bir RPC isteğinin başlık kısmına tekil bir işlem tanımlayıcı atanması ile bu sorun halledilmiştir.

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

- **CODA Mimari ve İletişim:**
 - CODA, NFS'nin aksine yüksek dereceli elde edilebilirliği ana amaç olarak belirlemiştir. Bu yüzden önbellekleme yapmış bir istemci için uygun şema sayesinde sunucu ile bağlantı kopmuş olsa bile ilgili istemci dosya üstünde değişiklik yapmaya devam edebilmektedir.
 - AFS versiyon 2'nin takipçisi olarak CODA geliştirilmiştir. Bu yüzden AFS'nin temel mimarisi hakkında fikir sahibi olmadan CODA'yı anlamak mümkün olmaz.
 - CODA, AFS ile aynı organizasyona sahiptir.

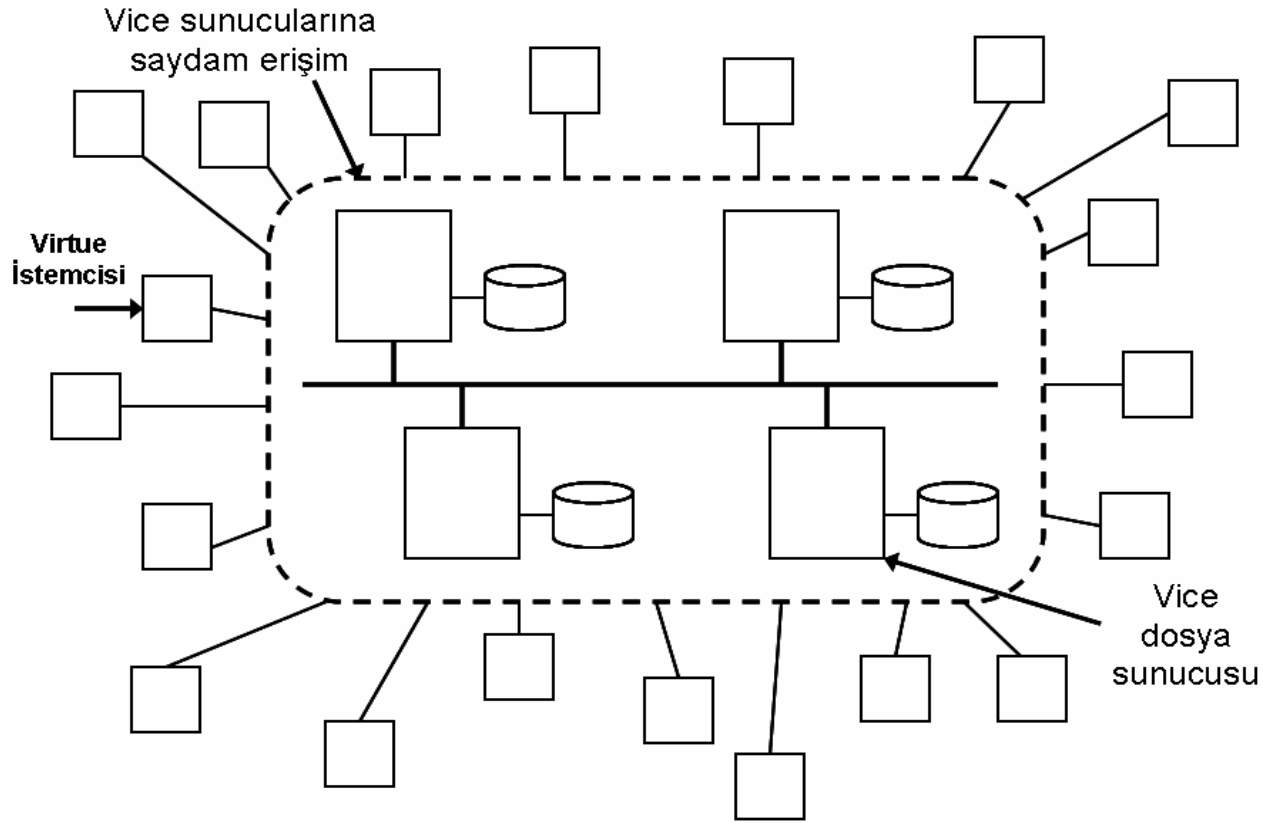
En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

● **CODA Mimari ve İletişim:**

- Her bir Virtue iş istasyonu makinesi Venus isimli kullanıcı-seviyesi sürece sahiptir ve bunun görevi NFS'deki istemciye benzetmektedir.
- Venus süreci, Vice dosya sunucusu tarafından bulundurulmuş dosyalara erişimi sağlamaktan sorumludur.
- Venus, dosya sunucusuna erişimin mümkün olmadığı durumda da istemcinin işlemlerine devam edebilmesine olanak tanır.

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

- **CODA Mimari ve İletişim:**



En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

- **CODA Mimari ve İletişim:**

- CODA'da RPC ile iletişim yerine RPC2 ile iletişim gerçekleştirilmektedir. RPC2'nin en önemli özelliği yan etkileri desteklemesidir.
- RPC2 multicasting'i de desteklemektedir. Bu özellik onu diğer RPC sistemlerden ayırır. RPC2 paketinin bir parçası olarak paralel (eşzamanlı) RPC mesajlaşması gerçekleştirilebilmektedir, buna **MultiRPC** denilmektedir.
- CODA için istemci süreçler Venus süreçleri, sunucu süreçler ise Vice süreçleri olarak kesinlikle ayrılmışlardır.

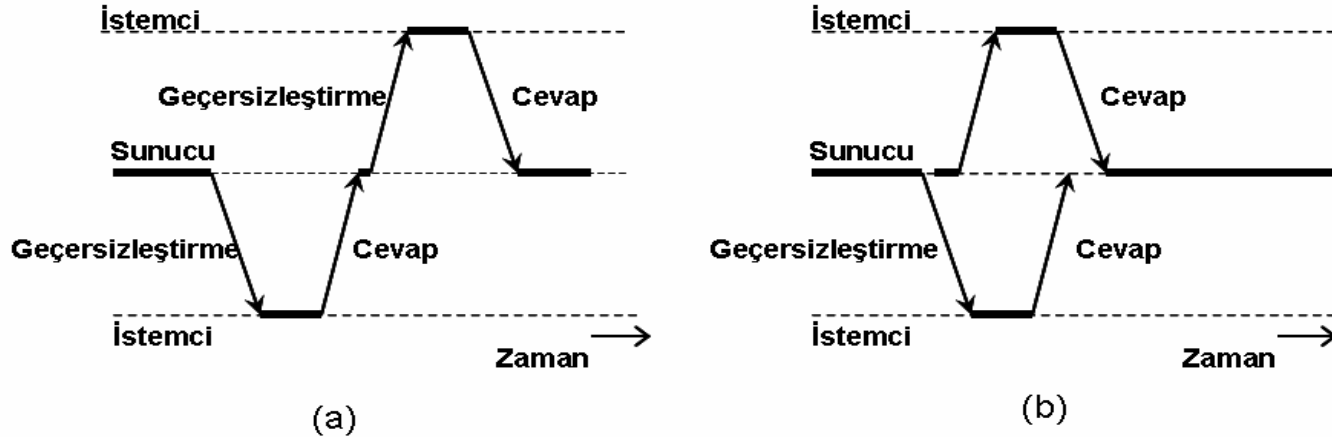
En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

● **CODA Mimari ve İletişim:**

- CODA sistemi sunucuları hangi istemcilerin bir dosyanın yerel kopyasına sahip olduğunu takip ederler.
- Bir dosya değişikliğe uğradığında, bir sunucu ilgili sistemleri RPC yoluyla uyararak yerel kopyaları geçersiz kılar.
- Eğer sunucu bir seferde sadece tek bir istemciyi uyarırsa, tüm istemcileri uyararak geçersizleştirme işlemi yapmak oldukça zaman alacaktır.
- Her bir kopyanın bire bir geçersizleştirilmesi yerine, sunucu tüm istemcilere bu mesajı paralel olarak göndererek ulaşılabilen durumdaki tüm istemcileri aynı zamanda değiştirmiş olur.
- Bu işlemde RPC mesajlaşma kullanıldığı için sunucu genel bir sona erme süresi içerisindeki RPC ile ulaşılamayan istemcileri çökmüş (ulaşamaz) olarak ilan eder.

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

- **CODA Mimari ve İletişim:**



RPC2 'deki geçersizleştirme mesajlaşması,
(a) bir seferde bir mesaj, (b) paralel mesaj gönderimi.

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

- **CODA İsimlendirme ve Senkronizasyon:**

- CODA, UNIX benzeri bir isimlendirme sistemi bulundurur. Dosyalar gruplanarak hacimlere ayrılmıştır. Bu hacimler UNIX'teki disk bölümlerine benzemektedir ve onlar gibi mount (bağlama) edilebilirler.
- Hacimler CODA için önemlidir, çünkü tüm isim uzayının oluşturulduğu temel birimlerdir. Bu yapı bağlanma noktalarına hacimlerin bağlanması şeklinde oluşturulur.
- İsimlendirme saydamlığının yüksek dereceden sağlanması için Vice dosya sunucusunun Venus sürecine isim look-up işlemi boyunca bağlama bilgisini geri döndürmesi gerekir.
- Bu bilgi sayesinde Venus gerektiğinde istemcinin isim uzayına bir hacmi otomatik olarak bağlayabilir. Bu mekanizma NFS v4'üne benzemektedir.

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

- **CODA İsimlendirme ve Senkronizasyon:**
 - Replikasyon açısından, her bir dosyanın tekil olarak tanımlanması önemlidir.
 - CODA'daki her dosya tam olarak bir hacim tarafından içerilmektedir.
 - Hacimler sunucular arasında replika edilebilirler. Bu yüzden CODA'da mantıksal ve fiziksel hacimler arasında bir ayırım yapılmıştır.
 - CODA, işlemsel semantik sağlamaya çalışır.

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

- **CODA İsimlendirme ve Senkronizasyon:**
 - CODA'nın asıl uğraştığı problem, oldukça büyük dağıtık dosya sisteminde, dosya sunucularının bazılarının veya tümünün geçici olarak ulaşılamaz olduğu durumdur.
 - Bu ulaşılamama durumu:
 - Sunucu veya ağda karşılaşılan bir hata olabileceği gibi,
 - Ağ servisinden kasten çıkan (bağlantısını koparan) bir mobil istemcinin neden olduğu durum da olabilmektedir.

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

- **CODA İsimlendirme ve Senkronizasyon:**
 - İstemci-tarafli önbellekleme CODA için iki sebepten dolayı önemlidir:
 - İlk sebep, AFS'de de izlenen yaklaşımda olduğu gibi ölçeklenebilirliğin başarılabilmesi için önbellekleme yapılmasıdır **[13]**.
 - İkinci sebep, istemciyi sunucunun elde edilebilirliğine daha az bağımlı hale getiren hataya dayanıklılığın yüksek derecesinin başarılmasını sağlayan konunun önbellekleme oluşudur **[14]**.
 - Bu sebeplerden dolayı CODA'daki istemciler tüm dosyaları önbelleklerle.

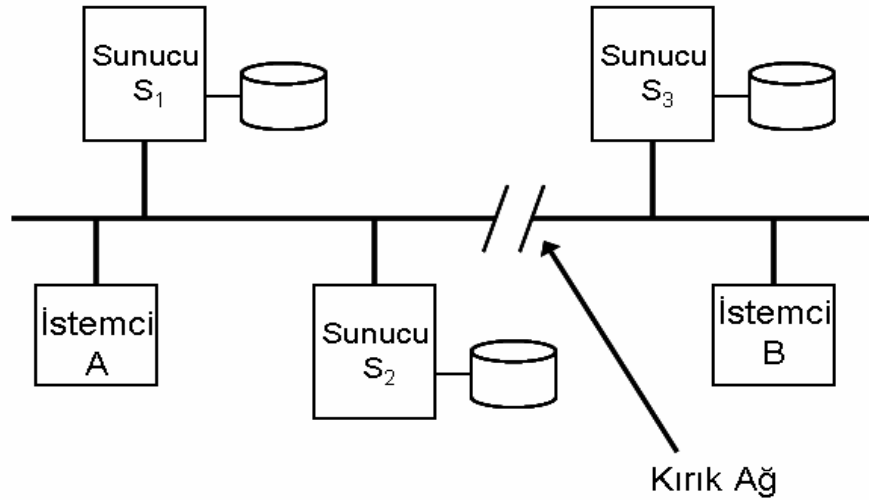
En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

● **CODA İsimlendirme ve Senkronizasyon:**

- CODA'da önbellek uyumluluğu geri döndürülen çağrılarla (callback) sağlanır.
- CODA, dosya sunucularının replika edilmesine izin verir.
- Bir hacmin bir kopyasına sahip sunuculardan oluşan topluluğa hacmin Hacim Depolama Grubu (VSG) denilir.
- Hata oluştuğunda, istemci hacmin VSG'sindeki tüm sunuculara erişemeyebilir.
- Bir istemci'nin Erişilebilir Hacim Depolama Grubu (AVSG), istemcinin iletişimde olduğu sunuculardan (VSG'deki) oluşur. Eğer AVSG boş ise, istemcinin bağlantısını kopardığı söylenilebilir.

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

- **CODA İsimlendirme ve Senkronizasyon:**



Replika edilmiş aynı dosya için farklı birer AVSG'ye sahip iki istemcinin durumu.

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

- **CODA İsimlendirme ve Senkronizasyon:**
 - CODA'da replikasyon stratejisi olarak, ***Read-One – Write-All (ROWA)*** kullanılmaktadır.
 - Bu yaklaşım, kopyalamanın yapıldığı tüm replikalarda şu anki verinin hepsinin bulunabilmesi olasılığını artırır.

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

● **CODA Hataya karşı dayanıklılık:**

- CODA'da, Sunucu replikasyonun aksine bağlantı kopartma esnekliği replikasyon protokollerinin performans kaybı olmadan sağlanılmaktadır.
- Bağlantı koparma işlemi bittikten sonra, bağlantısı kopmuş hacimlerden değişikliğe uğramış dosyalar ve dizinler AVSG'ye doğru yayımlanılır.
- Bir istemci kasten ağdan bağlantısını kopartarak, bağlantı kopartma işlemine gönüllü olarak girebilir.
- Kullanıcı büyük bir disk önbelleğine sahipse, belirli bir süre boyunca CODA sunucusundan izole olarak çalışabilir. Dosya isim uzayını kullanıcı değişmemiş olarak görür.
- Zaman zaman, kullanıcı ağdan çıkıp tekrar girmek isteyebilir, bu zamanlarda yaptığı değişiklikleri CODA sunucularına yayımlaması gerekir.
- CODA kablosuz teknolojileri de desteklemektedir.

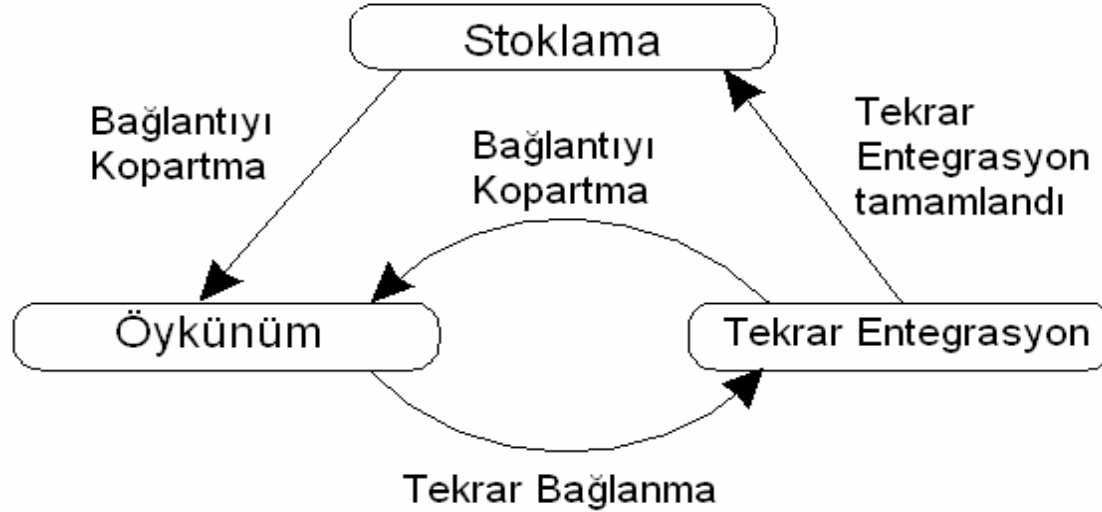
En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

● **CODA Hataya karşı dayanıklılık:**

- CODA'da, CODA istemcileri stoklama (hoarding) denilen bir strateji kullanır, buna göre istemciler hala bağlıyken önceden önbellek doldurulur.
- Stoklama fazı boyunca, istemci sürekli kendi önbelleğini güncellemek için çalışır.
- Bağlantı koparılması üzerine, istemci *öykünüm* (emulation) fazına girer. Bu durumda tüm işlemler yerel önbelleklenmiş kopyalar üzerinden yapılır.

En yaygın olarak kullanılan Dağıtık Dosya Sistemleri'nin detayları

- **CODA Hataya karşı dayanıklılık:**



Hataya dayanıklılık için CODA'daki Stoklama (Hoarding) stratejisi.

Modern Dağıtık Dosya Sistemlerinin Karşılaştırılması

İsimplendirme ve Senkronizasyon

- İsim organizasyonunda temelde iki yaklaşım bulunur:
 - İlk yaklaşımda, (NFS ve Plan 9'da izlenen) her bir kullanıcının kendi özel isim uzayını almasıdır. Bunun tek eksiği, isim tabanlı olarak dosya paylaşımının zor hale gelmesidir.
 - İkinci yaklaşımda ise, bir global paylaşımolu isim uzayı bulunur (CODA ve XFS'teki gibi). Tüm böyle sistemlerde, her bir kullanıcı özel bir yerel isim uzayı ile global isim uzayına eklenti yapabilme yeteneğine sahiptir.
- GoogleFS de yerleşimden bağımsız bir isim uzayını ve yük dengesi-hataya dayanıklılık için verinin saydam olarak taşınabilmesine olanak sunar.
- NFS oturum semantik'i sağladığı için, sunucu tarafından sadece bir dosyayı kapatan en son sürecin yaptığı güncelleme hatırlanmaktadır.
- CODA'da, işlemsel (transactional) semantik bulunur.

Modern Dağıtık Dosya Sistemlerinin Karşılaştırılması

Önbellekleme ve Hataya Dayanıklılık

- Sadece Plan 9, yazma işlemine dayalı önbellek tutarlılık protokolünü kullanır. Bu sayede her bir yazma işlemi hemen sunucuya iletilmiş olur.
- NFS v4'te bir dosya kapatıldığında değişiklikler sunuculara yayımlanacak şekilde düzenleme yapılmıştır.
- XFS dosya veri bloklarını önbellekler.
- CODA, tüm dosyanın önbelleklenmesini destekler.
- AFS'nin aksine, GoogleFS bir dosya verisini depolama sunucularına XFS gibi dağıtır.
- MS-DFS, isim uzayında GoogleFS gibi yerleşim bağımsız bir yapı sergilerken bağlanma büyüklüğü dizin seviyesinde bulunur ve RPC iletişim yapar.
- GoogleFS, HeartBeat (Kalp atışı) denilen bir mesaj türü kullanmaktadır.
- MS-DFS, replikasyon (sunucuları kopyalama) için MS-FRS denilen altyapıyı kullanılmaktadır.

Güvenlik

- NFS v4'te güvenli kanalların gerçekleştirimi için standart bir arayüz vardır.
- CODA ve Plan 9 güvenli kanalları sağlarlar.
- XFS tarafından güvenli kanallar sağlanmaz. Güvenilmeyen makinelerin XFS sistemine erişimine izin vermek için özel ölçütler belirlenmelidir
- NFS v4'te kapsamlı bir işlemler listesi erişim kontrolü için kullanılır.

Yukarıda bahsedilen isimlendirme, önbellekleme ve güvenlik kriterlerine göre çalışmamızda bahsedilen dağıtık dosya sistemlerini karşılaştırdığımız tabloya bakalım.



Modern Dağıtık Dosya Sistemlerinin Karşılaştırılması

Konu	NFS	CODA	Plan 9	XFS	OpenAFS	GoogleFS	MS-DFS
Tasarım amacı	Erişim saydamlığı	Yüksek elde edilebilirlik	Tekdüzelik	Sunucusuz sistem	Yüksek elde edilebilirlik	Veri depolama ve kümeleme	Yüksek elde edilebilirlik
Erişim modeli	Uzak	Upload/Download	Uzak	Log-tabanlı	Upload/Download	Seri numara ile kiralama	Uzak
İletişim	RPC	RPC	Özel	Aktif mesaj	RPC	Kalp atışı mesajı	RPC
Bağlanma büyüklüğü	Dizin	Dosya sistemi	Dosya sistemi	Dosya sistemi	Dosya sistemi	Chunk	Dizin
İsim uzayı	İstemci başına	Global	Süreç başına	Global	Global	Yerleşim bağımsız	Yerleşim bağımsız
Replikasyon	Minimal	ROWA	Yok	Striping	ROWA	Master veya chunk replikasyonu	FRS
Hataya dayanıklılık	Güvenli iletişim	Replikasyon ve önbellekleme	Güvenli iletişim	Striping	Replikasyon ve önbellekleme	Hızlı kurtarma ve replikasyon	Replikasyon
Paylaşım semantik	Oturum	İşlemsel	UNIX	UNIX	İşlemsel	UNIX-POSIX benzeri	UNIX benzeri
Önbellekleme birimi	Dosya (NFS v4' te)	Dosya	Dosya	Blok	Dosya	Chunk	Dosya
Erişim kontrolü	İşlemler ile	Dizin işlemleri	UNIX tabanlı	UNIX tabanlı	Dizin işlemleri	UNIX tabanlı	İşlemler ile

Tabloda kullanılan bazı kısaltmalar:

FRS : Dosya Replikasyon Servisi, **ROWA** : Read-one, Write-all yaklaşımı, **RPC** : Uzak Yordam Çağrısı.

SONUÇLAR

- Günümüzde dağıtık dosya sistemleri, küçük ölçekli projelerden başlayarak, büyük çaplı ve görece büyük kapasite isterlerine sahip projelerde müşteriler veya istemcilerin güvenlik/performans ve yüksek elde edilebilirlik ve hataya dayanıklılık isterlerini karşılamakta yoğun olarak kullanılmaktadır.
- Erişim konusunda farklı modeller kullanan birçok dağıtık dosya sistemi, hataya karşı dayanıklılık konusunda genel olarak güvenli iletişim kanalları, replikasyon ve önbellekleme'yi kullanmaktadırlar.
- Önbellekleme, sunucuya bağımlılığın azalarak daha fazla hataya dayanıklı bir yapı oluşmasına yardımcı olur.

SONUÇLAR

- Geleceğin dağıtık dosya sistemleri, mobilitenin daha da önem kazandığı, sunucusuz sistemlerin, yönetsel alanlar arasında ortak veri bloklarının serbest dolaşımına izin veren temel yapı üzerine kurulacaktır.
- Bu yapının esnekliği, onun istemci/sunucu bütünden oluşan birimlerinin kendi verilerine uygulayabildiği işlemlerin çeşitliliğine ve ağ iletişimindeki kontrolün ne derece iyi olduğuna bağlı olacaktır.

Kaynakça

- [1] Svobodova, L., (1984), "File servers for network-based distributed systems", ACM Computing Surveys, 16(4):353-398, Dec. 1984.
- [2] Swinehart, D., McDaniel, G., Boggs, D., (1979), "WFS: A simple shared file system for a distributed environment", In Proc. of the 7th ACM Symp. on Oper. Syst. Princ., pages 9-17, Dec. 1979.
- [3]. CODA (CMU) websitesi, (2007), (Çevrimiçi:<http://www.coda.cs.cmu.edu/ljpaper/lj.html>)
- [4]. Wikipedia.org web sitesi - bazı dosya sistemleri, (2007), (Çevrimiçi: http://en.wikipedia.org/wiki/List_of_file_systems)
- [5]. Tannenbaum, S. A., Van Steen, M., (2002), "Distributed Systems Principles and Paradigms", Prentice Hall, ISBN-0-13-088893-1
- [6]. Callaghan, B., (2000), "*NFS Illustrated*", Reading, MA: Addison Wesley 2000, Cited on pages: 192,576,587.
- [7]. Kon, F., (1996), "Distributed File Systems Past, Present and Future A Distributed File System for 2006", Tech. Report, University of Illinois at Urbana-Champaign, 1996.
- [8]. Srinivasan, R., (1995), "RPC: Remote Procedure Call Protocol Specification Version 2", RFC 1831, Aug. 1995, Cited on page 581.
- [9]. Bloomer, J., (1992), "Power Programming with RPC", Sebastopol, CA, O'Reilly & Associates, Cited on page 581.
- [10]. Juszczak, C., (1990), "Improving the Performance and Correctness of an NFS Server", Proc. Summer Techn. Conf., USENIX, pp53-63, Cited on page 598.

Kaynakça

- [11]. Howard, J., vd., (1988), "Scale and Performance in a Distributed File System", ACM Trans. Comp. Syst. , vol. 6, no. 2, pp 55-81, Feb. 1988, Cited on page 604.
- [12]. Satyanarayanan, M., Kistler, J., vd. , (1990), "Coda : A Highly Available File System for a Distributed Workstation Environment", IEEE Trans. Comp., vol. 39, no. 4, pp 447-459, Apr. 1990, Cited on page 604.
- [13]. Satyanarayanan, M., Siegel, E., (1990), "Parallel Communication in a Large Distributed System", IEEE Trans. Comp., vol. 39, no. 3, pp 328-348, Mar 1990, Cited on Page 608.
- [14]. Satyanarayanan, M., (1990), "Scalable, Secure and Highly Available Distributed File Access", IEEE Computer, vol. 23, no. 5, pp 9-21, May 1990, Cited on page 615.
- [15]. Satyanarayanan, M., (1992), "The Influence of Scale on Distributed File System Design", IEEE Trans. Softw. Eng., vol. 18, no. 1, pp 1-8, Jan. 1992, Cited on Page 615.
- [16]. Lampson, B., Abadi, M., Burrows, M., Wobber, E., (1991), "Authentication in Distributed Systems: Theory and Practice", Proc. 13th ACM Symp. on Op. Sys. Princ., Asilomar, 1991, pp. 165-182.
- [17]. Pike, R., Presotto, D., Dorward, S., Flandrena, B., vd., (1995), "Plan 9 from Bell Labs", Comp. Syst., vol. 8, no. 3, pp 221-254, Summer 1995, Cited on Pages 38, 188 ,624.
- [18]. Bell Labs Comp. Sci. Research Center, (2000), "Plan 9 programmer's Manual", Vol. 1., Bell Labs, Lucent Technologies, Murray Hill, NJ, 3rd ed., 2000, Cited on page 624.

Kaynakça

- [19]. Anderson, T., Dahlin, M., Neefe, J., Rosell, D., vd., (1996), "Serverless Network File Systems", ACM Trans. Comp. Syst., vol. 14, no. 1, pp 41-79, Feb 1996, Cited on pages 629, 630, 634.
- [20]. Anderson, T., Culler, D., Patterson, D., The NOW Team, (1995), "A Case for NOW", IEEE Micro, vol. 15, no. 2, pp 54-64, Feb. 1995, Cited on pages 629, 633.
- [21]. OpenAFS web sitesi. (Çevrimiçi : <http://www.openafs.org/>)
- [22]. Ghemawat, S., Gobioff, H., Leung, S., (2003), "The Google File System", 19th ACM Symp. on Oper. Syst. Princ., Lake George, NY, Oct., 2003, (Çevrimiçi:<http://labs.google.com/papers/gfs.html>)
- [23]. Arpaci-Dusseau, R.H., Anderson, E., Treuhaft, N., vd., (1999), "Cluster I/O with River: Making the fast case common", In Proc. of the Sixth Wrkshp on I/O in Paral.&Dist. Syst. (IOPADS '99), pages 10–22, Atlanta, Georgia, May 1999.
- [24]. Microsoft DFS Resmi Web Sitesi, (2007), (Çevrimiçi: <http://www.microsoft.com/windowsserver2003/technologies/storage/dfs/default.msp>)
- [25]. Microsoft DFS – Wikipedia.org Web Sitesi, (2007), (Çevrimiçi: http://en.wikipedia.org/wiki/Distributed_File_System_%28Microsoft%29)

Teşekkürler...

- Dinlediğiniz için teşekkürler...
- Sorular ve iletişim :
bahadir.karasulu@ege.edu.tr

Akademik Bilişim 2008 Konferansı
30 Ocak – 01 Şubat 2008