**DAĞITIK DOSYA SİSTEMLERİ(DISTRIBUTED FILE SYSTEM)**

-Birden fazla bilgisayar ve sunucu üzerinden veri depolama anlamına gelir.Birden fazla sunucuyu tek bir sunucu gibi kullanmamıza olanak sağlar.Ağ tabanlı dosya sistemidir.Ağ üzerinden bağlantı sağlayan bilgisayarların sunucu üzerinde ki bilgilere sanki kendi dosyası gibi erişimini sağlar.

-Günümüzde dağıtık dosya sistemleri;küçük ölçekli projelerden,büyük çaplı projelere kadar müşteriler veya istemcilerin güvenlik/performans,yüksek elde edilebilirlik ve hataya karşı dayanlıklık isteklerini karşılamaktadır.Ayrıca önbellekleme ve sunucuya bağımlılığı azaltmıştır.

-Örnek olarak;NFS,CODA,Plan9,XFS,OpenAFS,GoogleFS,MS-DFS verilebilir.Hepsinin kendine göre avantajları vardır.

**Network File System(NFS):**Temelinde TCP/IP modelini uygulama kısmında yer alan bir protokoldür.Uzak dosya servisi ile çalışır.Bu model ile istemci uzaktaki bir sunucu tarafından yönetilen dosya sistemine saydam bir şekilde erişim sağlar.Sisteme eklenen dosyadan istemcinin haberi olmaz.

-İstemci ile sunucu arasında ki iletişim RPC protokolü ile sağlanmaktadır.

-Durumsuz sunucuyu desteklediği için;RPC iletişim aşamasında sorun yaşanabilir.Bu sorun ise isteklerin başlık kısmına tekil bir işlem tanımlayıcı atanması ile halledilmiştir.

NFS, 4 ayrı protokolün birleşmesi sonucunda oluşur.

**NFS:** Dosyaları oluşturma,arama,okuma ve yazmaya izin verir.Dosya istatistiğini ve güvenilirliğini de denetler.Arabirimi “nfsd” dir.

**Mountd:** Ağ üzerinde ki bir dosyaya ya da başka bir kaynağa erişim sağlamak için bu protokol kullanılarak bağlanılır.Mount ya da unmount talebi alan sunucu dosya dizilerine ait olan bilgileri tutmak zorundadır.Arabirimi “mountd” dir.

**NSM(Ağ Durum Ekranı):** Bir makinenin durumunu ağ düğümlerini izleyerek kontrol eder.Reboot eden makineyi haber verir.Arabirimi “statd”dir.

**NLM(Ağ Kilit Yöneticisi):** Aynı anda farklı sunucu/istemciler tarafından yapılan veri değiştirme isteklerini engellemek amacıyla bu protokol kullanılır.NSM protokolü yardımıyla da bir istemcinin ne zaman yeniden başladığını belirler.Arabirimi “lockd”dir.

**Ağ Üzerinde Paylaşım:** Yerel bir diskin mount edilmesi kadar kolaydır.

NFS için mount programının kullanımı şu şekildedir:

“#mount –t nfs <sunucu-adresi:paylaşma-dizini> <yerel-dizin> -o <seçenekler>”

**Avantajları:**

-Bir kullanıcı herhangi bir makinaya girip diğer kullanıcılar ile aynı dosya yapısı ile çalışabilir.

-Paylaşım sayesinde büyük disk alanı isteyen programlar tek bir sunucuda toplanabilir ve diskten tasarruf edilebilir.

-Bir makine üzerinde yapılan değişiklik her makinede yapılmış gibi olur.Aynı yazılımı 20 PC’ye yüklemek zorunda kalmazsınız.

**CODA File System:** (AFS)Andrew dosya sisteminin eski bir sürümünden türetilmiştir ve birçok benzer özellik sunar. NFS’nin aksine elde edilebilirliği amaçlamıştır.Yüksek derecede ölçeklenebilirlik ve güvenlik amaçlanmıştır.

-Coda,ağ bağlantısı kesildiğinde sunucu verilerine erişim sağlamak için yerel bir önbellek kullanır.İstemci ağ bağlantısının kesilmesi sırasında önemli olarak listelediği tüm verileri biriktirir ve yeniden ağa bağlanma durumunda hepsini sunuculara kayıt eder.

-AFS’den farklı olan yönü Coda’nın veri çoğaltma yöntemidir.AFS çoğaltma işlemi sırasında diğer sunuculara “salt okunur” biçimde gönderir ve üzerinde işlem yapmasını engeller.Coda da ise bu durum söz konusu değildir.

-RPC ile iletişim yerine RPC2 ile iletişim gerçekleştirmiştir.Multicasting desteği vardır.

-Dosyalar gruplanarak hacimlere ayrılmıştır.UNİX’teki disk bölümlerine benzer.

-CODA’daki istemciler tüm dosyalar önbelleklerler.Dosya okumak veya yazılmak içi açıldığında bir kopyası istemciye transfer edilip orada saklanır.

**Avantajları :**

-Mobil bilgi işlem için bağlantısız işlem.

-İstemci tarafında kalıcı önbelleğe alma yoluyla yüksek performans

-Sunucu çoğaltma

-Kimlik doğrulama, şifreleme ve erişim kontrolü için güvenlik modeli

-Sunucu ağındaki kısmi ağ arızalarında çalışmaya devam etme

-Ağ bant genişliği uyarlaması

-İyi ölçeklenebilirlik.

**Plan 9 File System :** 9P olarak da bilinir.Bell Labs dağıtılmış işletim sisteminden Plan9 için geliştirilmiş bir ağ protokolüdür.OS X ve Linux ta başarılı bir şekilde çalışır.OS X ve Linux hala hafif bir uzak dosya sistemi gerektiğinde bu protokolü kullanır.

-Dosya tabanlı bir dağıtık sistem değil bir dağıtık dosya sistemidir.

-İstemci sunucu ayrımı tam olarak yapılamamıştır.

-Ağ üzerinden iletişim 9P denilen standart bir protokolle sağlanır.

**Fonksiyonlar:**

**Version :** Sürüm.

**Error :** Hata.

**Flush :** Mesajı iptal etme.

**Auth-Attach :** Bağlantı kurma ve yetkilendirme.

**Walk :** Hiyerarşiler arası geçiş.

**Create-Open :** Mevcut dosya oluşturma veya açma

**Read-Write :** Dosyadan ve dosyaya veri aktarma.

**Remove :** Sunucudan dosya kaldırma.

**Stat-Wstat :** Dosya özniteliklerini açma ve değiştirme.

**XFS:** Yüksek performanslı 64-bit günlük dosya kayıt sistemidir.Linux tarafından desteklenmektedir.Giriş/Çıkış(I/O) tasarımına dayanmaktadır.Asıl amacı yüksek dereceden derecelendirilebilirlik ve hataya dayanıklık sağlamaktır.

-Önceden RPC kullanılırken performans kaybı nedeniyle aktif mesaj yaklaşımına geçilmiştir.

-Sunucusuz tasarıma sahiptir birden çok makine arasına dağıtılmıştır.AFS ve CODA merkezi bir organizasyonken XFS değildir.

-Alan tahsisi ağaç yapısında depolanan veri yapıları ile sağlanmaktadır.Delayed allocation ise dosya sisteminin parçalanmasının önlenmesini sağlar.

-Herhangi bir elektrik kesintisinde veya sistem çökmesine karşı verilerin tutarlılığını sağlar.Dosya sistemi meta verileri için günlük kayıt sağlar.

-Değişken blok boyutlarına sahiptir.512 bayt ile 64 KB arasında değişkenlik gösterir.Bu özellik yüksek boyutlu bloklarda performansı düşürür.

-Genişletilmiş öznitelikler bulundurur.Öznitelikleri “root” ve “user” olarak depolar.

-Hiyerarşik depolama yönetimi için DMAPI arabirimini uygular.

-XFS,anlık olarak görüntü işleyemez.Bunun için “xfs\_freeze” yardımcı programı kullanılır.

-Çevrimiçi birleştirme işlemi için “xfs\_fsr” kullanılır.

-Çevrimiçi bir şekilde yeniden boyutlandırma için “xfs\_growfs” kullanılır.

**Dezavantajları :**

-Meta verileri; diğer günlük kayıt dosya sistemlerine göre daha yavaş işleyebildi.

-Günlük kayıt ise devre dışı bırakılamadığı için SSD’lerin ömrü fazlasıyla kısaldı.

**OPEN AFS :** İstemci/Sunucu mantığıyla çalışır.Sunucu makinede,dosyalar yer alır ve istemci makineler bu dosyaları kendi makinesiymiş gibi kullanabilir.

-Kimlik doğrulama işlemi için Kerberos kullanır.Bu ise fazlasla güvenilir olmasını sağlıyor.

-Sunucu kümeleri oluşturma,Sunucu kapatıp açma,Çalışma anından “Back-up” işlemi yapabilme gibi işlemler sağlar.

-Cache mekanizması ile istediğimiz dosya,istemci cache’sinde varsa ve herhangi bir değişiklik yoksa doğrudan cache’den çalışmaya başlayabiliyoruz.Her seferin de sunucudan istemeye gerek kalmaz.

-Unix,Linux,MAC OS X ve Windows’ta hem sunucu hem istemci olarak çalışıyor.

**MS-DFS :** Microsoft firmasının geliştirdiği dağıtık dosya sistemidir.İstemci ve sunucu servislerinin bir kümesi olarak dağıtık dosya paylaşımlarını organize ederek bunu bir dağıtık dosya sistemine dönüştüren yapıdır.

-DFS;sınırlı ağ bağlantılarına sahip network üzerinde çalışan sunucular arasındaki dosya alışverişinin senkronizasyonunu sağlayan bir çoğaltma altyapısıdır.

-Uzaktan sıkıştırma olarak RDC algoritmasını kullanmaktadır.RDC;dosya da değişiklik olduğu zaman dosyanın tamamını değil değişiklik olan yerin kopyalanmasını sağlar.

-Ücretsiz olarak kullanım sağlanmamaktadır.

-Yerleşim saydamlığı,hatanın varlığı veya birden çok paylaşımın olduğu durumlarda ki yoğun yük altında veri elde edilebilirliğinin iyileştirilmesini sağlar.

-DFS ile birlikte Veri tabanı kurtarma özelliği de eklenmiştir.

**GoogleFS :**  Geniş ölçekli dağıtık loglanabilir ve kontol altında tutulabilir bir dosya sistemidir.

-Snapshot ve kayıt ekleme işlemlerini içerir.Aynı dosyaya eşzamanlı olarak veri eklenebilir.

-Standart bir API uygulanmaz diğer sistemlere benzer olarak hiyerarşik yapıda düzenlenmiş dizinler ve dosyalarla ilgilenir.Dosyalara “path” lar üzerinden erişim sağlanır.

**GFS Master :** Dosya yolu adlarını ve bu dosyalar ait metadaları tutar.İstemciler ile iletişimi sağlar.Metadata’ya bakarak hangi Chunk Server üzerinde tutulduğunu öğrenir.Yeni sunucu oluşturma,veriye erişim hızı ayarlanması,verilerin eşiğinin ayarlanması,namespace yönetimi,eskiyen kopyaların silinmesi işlemlerini gerçekleştirir.

**GFS Chunk Server :** Her Chunk Server;datayı 3 farklı CS’a çöktüğünde dataya erişim olması amacıyla replike eder.Datalar 64 MB sabit uzunlukludur.Network yükünü azaltır.Dezavantajı ise birden fazla client aynı dosyaya erişim sağlamak istediğinde chunk serverlar üzerinde hot spot oluşabilir.

**Metadata :** 3 Tip metadata bilgisli saklar.

-İlk 2 tip metadata loglama içinde kullanılır.

-Dosya adı ve bağlı olduğu CS adı.2

-Dosyalardan CS’lara haritalama.

-Her bir Chunk replikalarının yerleri.

**Okuma algoritması :**

-Uygulama,GFS istemciye okuma isteği oluşturur.

-GFS istemci,Master’e isteği iletir.

-Master,chunk ve replika lokasyonlarını döndürür.

-İstemci okuma yapacağı yeri seçer ve istek gönderir.

-Chunk sunucu datayı GFS istemciye gönderir.

-İstemci okuma yapacağı verileri uygulamaya iletir.

**Yazma Algoritması :**

-Uygulama istek oluşturur. GFS istemci , Master’e isteği iletir.

-Master, GFS istemciye birincil ve ikincil replika yerlerini yanıt olarak döner.

-İstemci tüm chunklara yazmak için veri gönderir ve Veriler CS içindeki bufferlarda saklanır.

-İstemci,birincil replikaya yazma komutu gönderir.Komutun gemesiyle data örneğini kendi bufferına alır ve chunka yazar.

-Birincil replika yazıldıktan sonra ikincil replikalara yazma işlemini yapılması için seri komutlar gönderir.

**Snapshot :** Sistemin o an için bir kopyasının oluşturulduğu anlık görüntüdür.Master “SS” isteği aldığı zaman sunucularda ki tamamlanmamış görevleri iptal ederek işlemi tamamlar.

**Replica Management :** Veri kullanılabilirliği ve güvenilirliği maksimize etmek için kullanılır.Bant genişliği kullanımını maksimize etmek için ise Chunk replikalarının makine ve crckler arasında dağıtımını sağlar.

**Çöp Toplama Mekanizması :**

-Dosya ilk olarak gizli dosya olarak işaretlenir.

-İşaretlenen bu dosya 3 gün geçtikten sonra silinir.

-Dosya silindikten sonra, bellekte ki metadatası da silinir.

-Düzenli olarak Chunk namespace i taranır ve sahipsiz chunklar da silinir.

**Shadow Masters :** Veri bütünlüğünü kontrol eder ve Her bir chunk için 64 KB sağlama bloğu vardır.

**Checksumming :** Her chunk sunucusu depolanan veriler de bozulmayı tespit etmek için kullanır.Checksumlar datadan ayrı olarak bellekte tutulur.Okuma hatası aldığında bunu master’a bildirir ve yeniden kopyalama ve karşılaştırma işlemleri gerçekleştirilir.

**Avantajları :**

-Snapshot(anlık) ve kayıt defterine ekleme işlemini gerçekleştirir.Snapshot düşük bir maliyetle bir dosyanın kopyası alır ve bir dizin ağacı oluşturur.

-Farklı “Datacenter”lar üzerinde çalışabilir.

-Binlerce makine aynı dataya erişim sağlayabilir(Read,Write).

-Yüksek boyutlu dosya destekler.

-Erişilen makineler üzerinde eşit yük dağılımı yapar ve dar boğazları engeller

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **KONU** | **NFS** | **CODA** | **PLAN9** | **MS-DFS** | **XFS** | **OPENAFS** | **GoogleFS** |
| **Tasarımın Amacı** | Erişim saydamlığı | Yüksek elde edilebilirlik | Tekdüzelik | Yüksek elde edilebilirlik | Sunucusuz sistem | Yüksek elde edilebilirlik | Veri depolama ve kümeleme |
| **Erişim**  **Modeli** | Uzak | Upload/Download | Uzak | Uzak | Log tabanlı | Upload/Download | Seri no ile kiralama |
| **İletişim** | RPC | RPC | Özel | RPC | Aktif Mesaj | RPC | Kalp atışı mesajı |
| **Bağlanma**  **Büyüklüğü** | Dizin | Dosya Sistemi | Dosya Sistemi | Dizin | Dosya Sistemi | Dosya Sistemi | Chunk |
| **İsim**  **Uzayı** | İstemci başına | Global | Süreç Başına | Yerleşim bağımsız | Global | Global | Yerleşim bağımsız |
| **Replikasyon** | Minimal | ROWA | Yok | FRS | Striping | Rowa | Master ve Chunk |
| **Hataya**  **Dayanıklılık** | Güvenli İletişim | Replikasyon ve Önbellekleme | Güvenli İletişim | Replikasyon | Striping | Replikasyon ve Önbellekleme | Hızlı kurtarma ve replikasyon |
| **Paylaşım**  **Semantik** | Oturum | İşlemsel | UNIX | UNIX benzeri | UNIX | İşlemsel | UNIX-POSIX Benzeri |
| **Önbellekleme**  **Birimi** | Dosya | Dosya | Dosya | Dosya | Blok | Dosya | Chunk |
| **Erişim**  **Kontrolü** | İşlemler ile | Dizin işlemleri | UNIX tabanlı | İşlemler ile | UNIX tabanlı | Dizin işlemleri | UNIX tabanlı |

**KARŞILAŞTIRMA TABLOSU**