

Güncel Dağıtık Dosya Sistemlerinin Karşılaştırmalı Analizi

Süleyman Eken
suleyman.eken@kocaeli.edu.tr

Kocaeli Üniversitesi
Bilgisayar Mühendisliği Bölümü

Dosya Nedir ?

Kalıcı depolama bilgisayarlarda kullanılan en temel soyutlamalardan biridir.

Buna dair temel problemleri çözmek için dosya adı verilen yapı kullanılır.

Bir dosya süreçler tarafında üretilen verilerin saklandığı mantıksal birimlerdir. Dosyalar hem verinin kendisini hem de veriye dair bilgileri içerir.

Dosya Sistemi Nedir ?

Dosya sistemleri dosyaların bulunması, depolanması, isimlendirilmesi, korunması, organizasyonu gibi işlerden sorumlu yazılımlardır.

Ayrıca programcılara dosyalarla çalışmak için programlama arayüzü sağlarlar.

Dağıtık Sistem ve Dağıtık Dosya Sistemi (DDS) ?

Bir dağıtık sistem kullanıcılara tek bir sistemmiş gibi görünen birden fazla bağımsız bilgisayarın toplanmasıyla oluşur.

Dağıtık dosya sistemleri için temel amaç farklı bilgisayarların dosyaları ve depolama kaynaklarını ortak kullanabilmelerini sağlamaktır.

Bu amaç gerçekleştirilmeye çalışılırken genelde UNIX dosya sistemi temel alınmaktadır.

DDS Gereksinimleri

- **Şeffaflık**

Kendini kullanıcılara ve uygulamalara tek bilgisayar gibi sunabilen dağıtık sistemlere şeffaf dağıtık sistemler denir.

- **Eşzamanlı Dosya Güncelleme**

Bir dosyada bir istemci tarafından yapılan değişiklikler diğer istemcilerin çalışmalarını ve dosyada değişiklik yapmalarını engellememelidir.

DDS Gereksinimleri

- **Dosya Replikasyonu**

Dosyalarının kopyalarının sistem boyunca dağıtılması sisteme güvenilirlik ve performans sağlayacaktır

- **Tutarlılık**

Dosyaların kopyalarının dağıtılması, kopyaların aynı olması sağlamak gibi bir gereklilik de ortaya çıkaracaktır.

DDS Gereksinimleri

- **Hata Toleransı**

Sistem bir hatayla karşılaştığında onarımlar yapılana kadar fazla performans kaybı olmadan çalışabilmelidir.

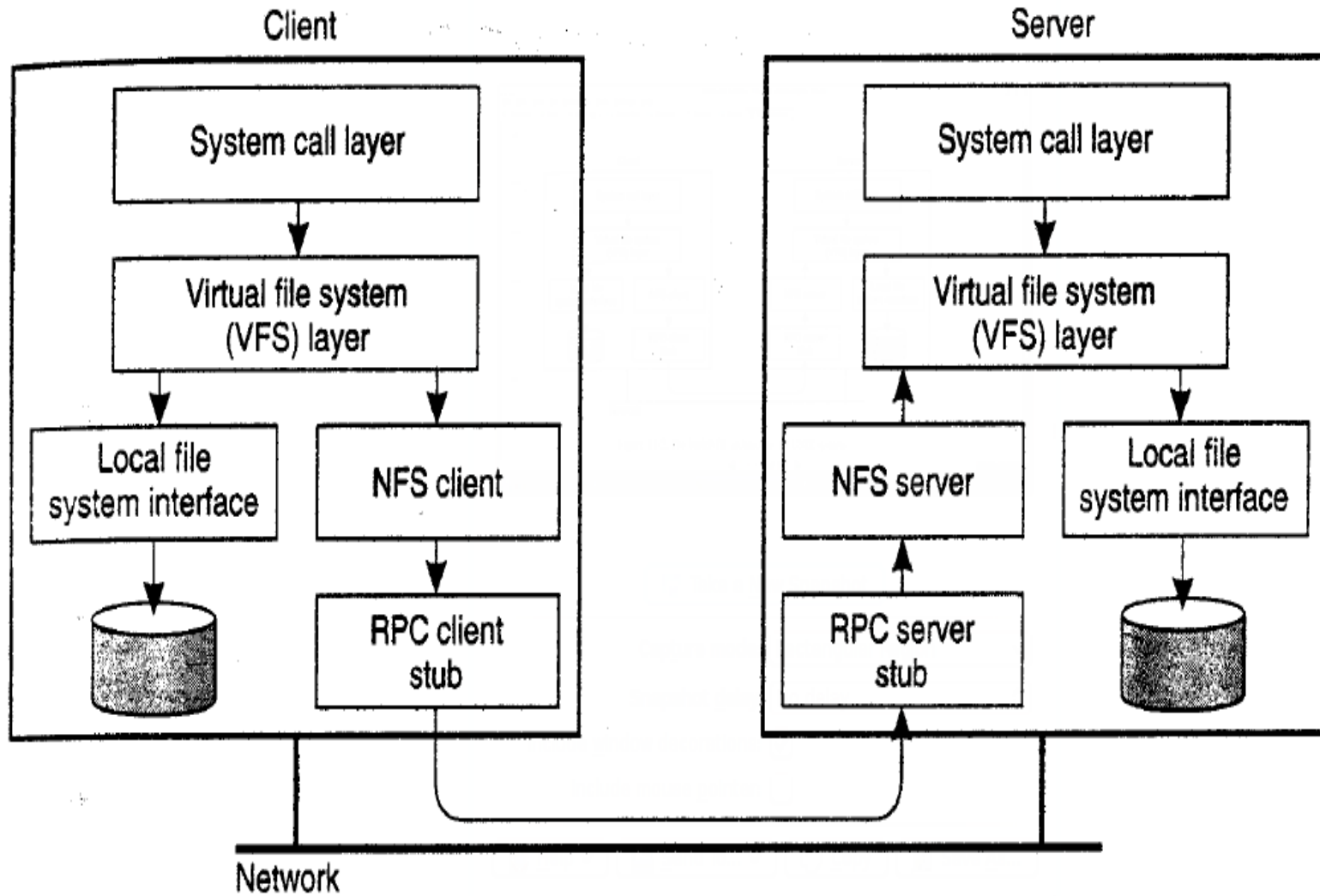
- Bunların yanı sıra bir DDS mümkün olduğu kadar güvenli, verimli ve platform bağımsız olmalıdır.

Örnek: Sun (Oracle) NFS (Ağ Dosya Sistemi)

NFS, Sun Microsystems tarafından ilk olarak 1984 yılında geliştirilmeye başlanmış hem bir dağıtık dosya sistemi protokolünün hem de dosya sisteminin kendisinin adıdır.

NFS'te genel tasarım amacı klasik UNIX dosya sisteminin dağıtık bir şekilde gerçekleştirilebilmesidir

NFS Mimarisi



NFS Özellikleri

- **İsimlendirme**

Standart UNIX bağlama işlemi kullanılır.
Dosya sistemlerinin tamamı veya bir kısmı bağlanabilir.

Dosya konumları → /etc/exports

- **İletişim**

Uzak metod çağrıları (RPC)

NFSv3 → Tekil NFSv4 → İteratif

NFS Özellikleri

- **Senkronizasyon**

Kilit mekanizması kullanılır.

- **Tutarlılık**

Önbellekteki dosya ve sunucudaki dosyanın zaman mühürleri karşılaştırılır.

- **Hata Toleransı**

NFSv3 → Durumsuz, NFSv4 → Durumlu sunucu modeli kullanır.

- **Güvenlik**

UNIX erişim kontrol sistemi kullanılır.

Örnek: AFS (Andrew Dosya Sistemi)

Andrew Dosya Sistemi 1983 yılında Carnegie Mellon Üniversitesi ve IBM işbirliği ile geliştirilmeye başlanmıştır. Tasarımın temel önceliği ölçeklenebilir olmasıdır.

AFS dizaynı dosya boyutları ve kullanımları üzerine bazı varsayımlardan etkilenmiştir.

AFS'te çalışan istemci yazılımları Venus, sunucu yazılımlarına ve bazen sunucuların tamamına Vice ismi verilir.

AFS Özellikleri

- **İsimlendirme**

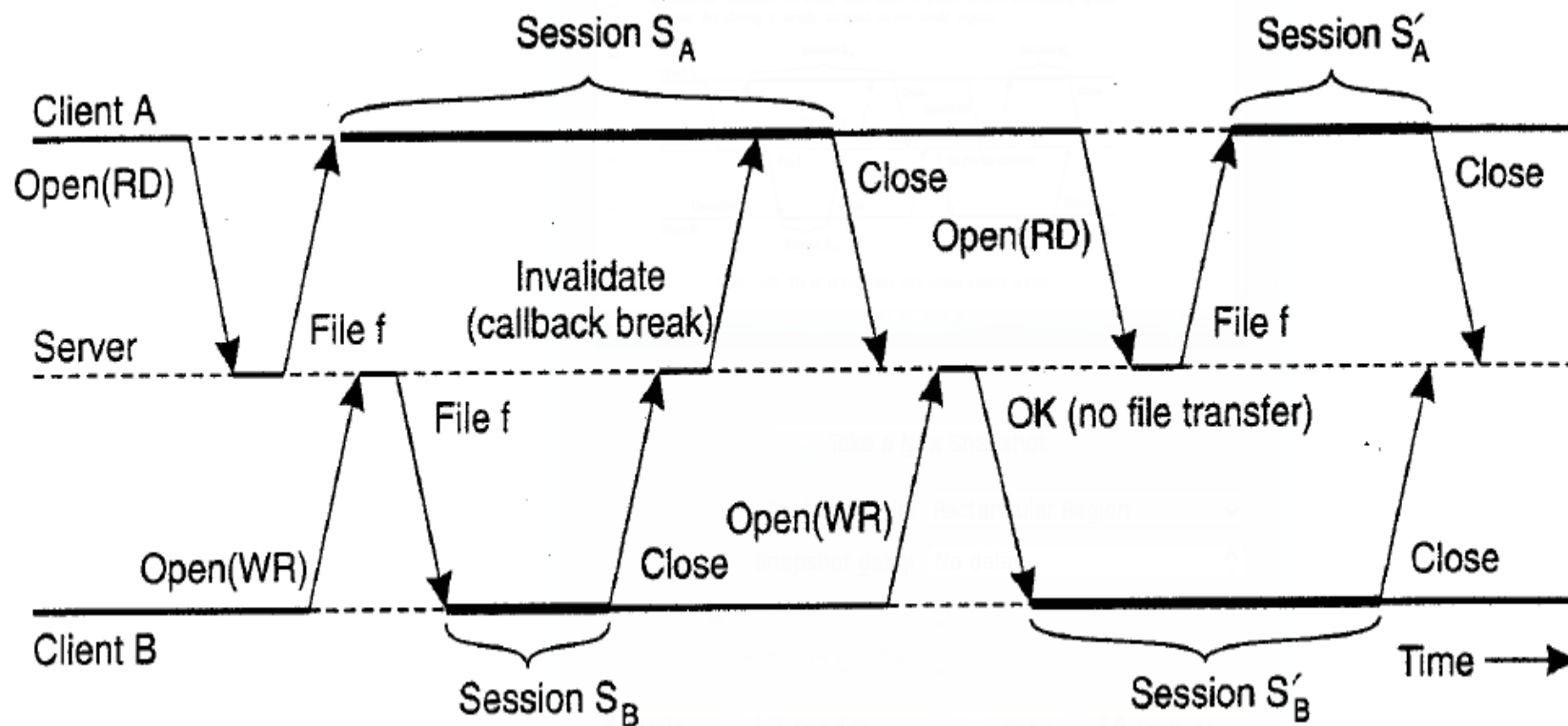
AFS dosya isim alanı ikiye ayrılır. Yerel ve paylaşılan, paylaşılan isim alanı şeffaf bir şekilde bütün kullanıcılarda aynıdır. Ayrıca hacim (volume) denilen mantıksal yapılar kullanılır.

- **Senkronizasyon**

AFS senkronizasyon için oturum semantiğini kullanmaktadır.

AFS Özellikleri

- Tutarlılık ve Replikasyon



AFS Özellikleri

- **Hata Toleransı**

AFS durumlu bir servis olduğu için hata toleransı NFS'e göre biraz daha düşüktür.

Bir istemci çöktüğünde callback jetonlarını çöpe atar ve sunucuya önbellekteki kopyalarının kontrol edilmesi için başvurur.

- **Güvenlik**

AFS'te güvenlik işlevi Vice sunucularına devredilmiştir. Bu sunucular fiziken güvenli kabul edilir ve üzerilerinde sadece güvenilen sistem yazılımları çalıştırılır. AFS güvenli RPC ve erişim kontrol mekanizmalarını içermektedir.

AFS Türevleri

- **Arla**

İsveç'te KTH Kraliyet Teknoloji Enstitüsü tarafından geliştirilen bir AFS istemcisidir. Arla'nın OpenAFS'ten temel farkı temel işlevlerin çekirdeğe yerleştirilmeyip kullanıcı seviyesinde daemonlar kullanılarak halledilmeleridir

- **Coda**

CMU'da AFS'i geliştiren Prof. Dr. Mahadev Satyanarayanan ve ekibinin AFS-2 tabanlı bir araştırma projesidir. AFS-2'den temel farkı sunucu tarafında replikasyon, çevrimdışı işlem gibi erişilebilirliği arttıran özellikler getirmesidir.

AFS Türevleri

- **OpenAFS**

1989 yılında CMU 'da çalışan bazı araştırmacılar AFS için kurumsal çözümler üretmek üzere Transarc isminde bir firma kurdular. Bu firma 1994 yılında IBM tarafından satın alındı. Transarc AFS 2000 yılına kadar IBM tarafından geliştirildikten sonra OpenAFS adıyla açık kaynaklı olarak yayınlandı. OpenAFS bir AFSv3 gerçeklemesi

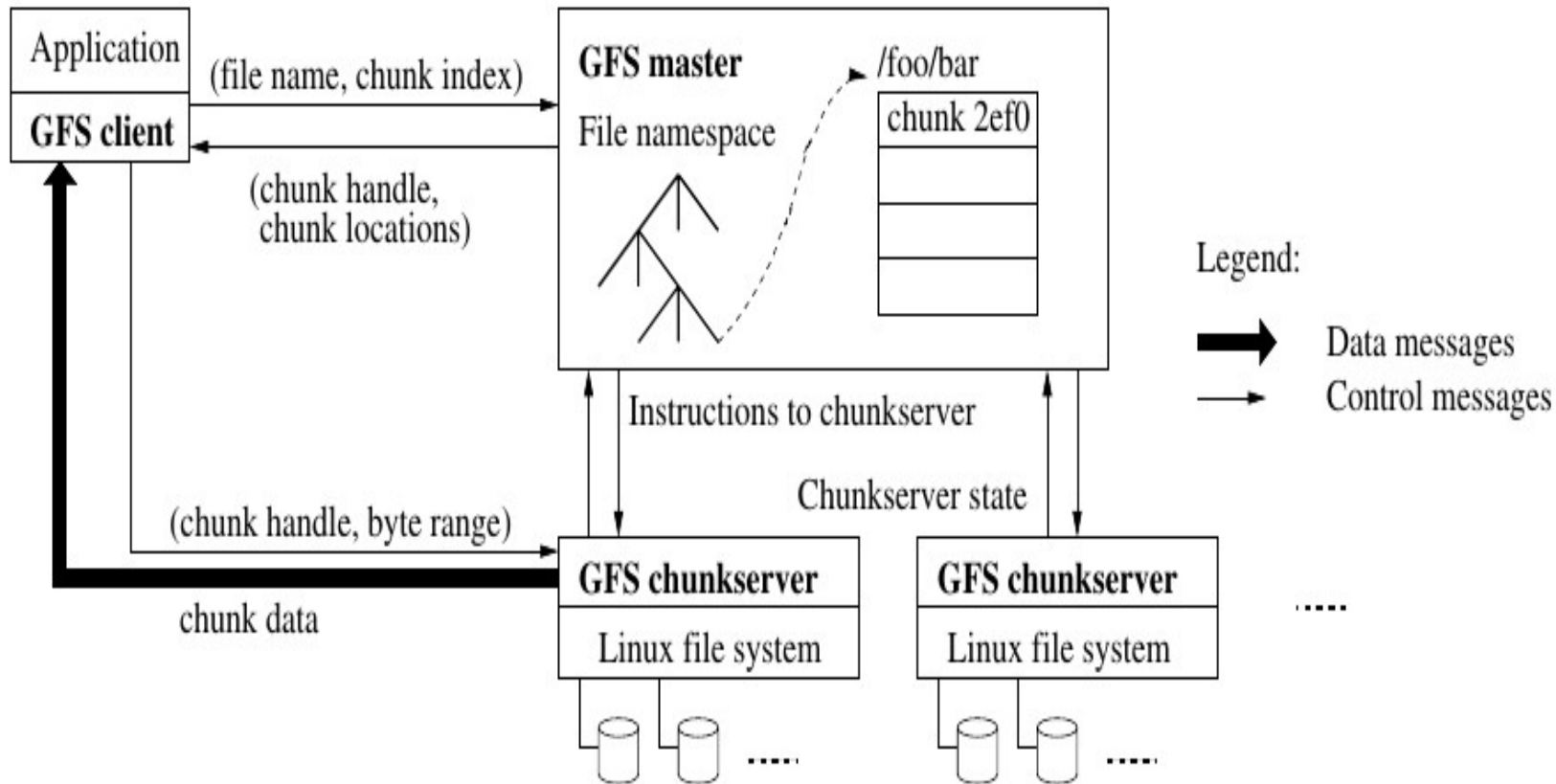
Örnek: GFS (Google Dosya Sistemi)

Google yaptığı iş gereği çok büyük verilerle çalışmaktadır.

Verileri yüksek kapasiteli az sayıda sunucuda depolamak ve işlemek oldukça maliyetli olmakta ve ölçeklenebilir olmamaktadır.

Çok miktarlarda ucuz donanım kullanıp sistemi hatalara karşı daha dayanıklı olması için tasarlamak daha uygun bir çözümdür.

GFS Mimarisi



GFS Özellikleri

- **İsimlendirme ve Şeffaflık**

GFS'te dizinler için ayrı bir veri yapısı bulunmaz. Ayrıca kısayol,bağ gibi yapılar da mevcut değildir. GFS isim alanını mantıksal olarak her yol isimlerini metadatalara eşleyen lookup tablolarıyla düzenler. Aynı zamanda kilitleme mekanizması mevcuttur.

- **Senkronizasyon**

GFS'te dosyalar iki türlü işlem ile değiştirilebilir. Yazma ve kayıt ekleme.

GFS Özellikleri

- **Tutarlılık ve Replikasyon**

GFS'te istemciler için önbellekler tutulmaz. Ancak her yığının kopyaları (ortalama 3 kopya) çeşitli sunucularda bulundurulur. Master'ın operasyon kayıtları ve checkpointleri çeşitli sunucularda bulundurulur.

- **Hata Toleransı**

Master ve yığın sunucular saniyeler içerisinde tekrar başlatılabilirler. Master tekrar başlatıldığında diskten operasyon kaydını okur.

GFS Türevleri

HDFS (Hadoop Dosya Sistemi) Apache Vakfı tarafından geliştirilen Hadoop projesinin bir parçasıdır.

HDFS, GFS'in açık kaynaklı bir gerçeklemesidir. Orjinal GFS C++ ile yazılmışken HDFS Java ile yazılmıştır.

GFS'teki Master, Chunk server, Operation Log, Shadow server gibi yapılar HDFS'te Namenode, Datanode, Edit log, Secondary namenode gibi isimler almışlardır.

HDFS'te GFS'ten farklı olarak kayıt ekleme, çöp toplama gibi mekanizmaları sağlayamamaktadır.

DDS'lerin Karşılaştırılması

Kriter	NFS	AFS		GFS
		CODA	OpenAFS	
Tasarım amacı	Erişim saydamlığı	Yüksek elde edilebilirlik	Yüksek elde edilebilirlik	Veri depolama ve kümeleme
Erişim modeli	Uzak	Upload/Download	Upload/Download	Seri numara ile kiralama
İletişim	UMÇ	UMÇ	UMÇ	Kalp atışı mesajı
Bağlanma büyüklüğü	Dizin	Dosya sistemi	Dosya sistemi	Chunk
İsim uzayı	İstemci başına	Global	Global	Yerleşim bağımsız
Replikasyon	Minimal	ROWA *	ROWA	Master veya chunk replikasyonu
Hataya dayanıklılık	Güvenli iletişim	Replikasyon ve önbellekleme	Replikasyon ve önbellekleme	Hızlı kurtarma ve replikasyon
Önbellekleme birimi	Dosya (NFS v4'te)	Dosya	Dosya	Chunk
Erişim kontrolü	İşlemler ile	Dizin işlemleri	Dizin işlemleri	UNIX tabanlı

*ROWA: Read-one, Write-all yaklaşımı.

**Dinlediğiniz İçin
Teşekkürler**