

# Distributed File Systems

---

PANJI WISNU WIRAWAN

# Agenda

---

Distributed File Systems

File Service Architecture

Studi Kasus: Sun Network File Systems

Studi Kasus: Hadoop Distributed File Systems

# Distributed File Systems

---

Tujuan system terdistribusi: resource sharing (storage, memory, .... )

Mekanisme data sharing: P2P file sharing, distributed storage (distributed file & object).

Tujuan distributed file service: menyimpan dan mengakses file secara remote.

File system: bertanggung jawab untuk organisasi, penyimpanan, naming, sharing, dan protection.

# Distributed File Systems

	<i>Sharing</i>	<i>Persis- tence</i>	<i>Distributed cache/replicas</i>	<i>Consistency maintenance</i>	<i>Example</i>
Main memory	×	×	×	1	RAM
File system	×	✓	×	1	UNIX file system
Distributed file system	✓	✓	✓	✓	Sun NFS
Web	✓	✓	✓	×	Web server
Distributed shared memory	✓	×	✓	✓	Ivy (DSM)
Remote objects (RMI/ORB)	✓	×	×	1	CORBA
Persistent object store	✓	✓	×	1	CORBA Persistent Object Service
Peer-to-peer storage system	✓	✓	✓	2	OceanStore

# Distributed File System:

## Kebutuhan

---

Transparency

Concurrent file updates

File Replication

Hardware & OS heterogeneity

Fault Tolerance

Consistency

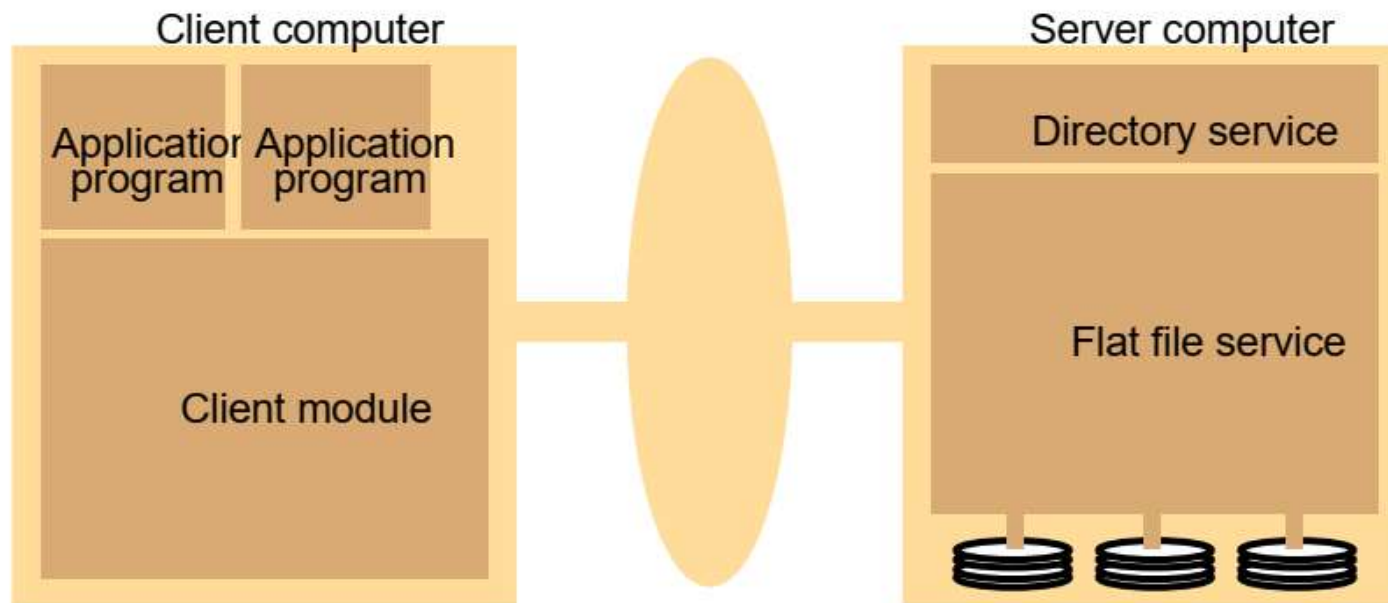
Security

Efficiency



# File Service Architecture

---



# File Service Architecture

---

Stateless file service architecture:

- Flat file service
- Directory service
- Client Module

# Studi Kasus: Sun Network File System

---

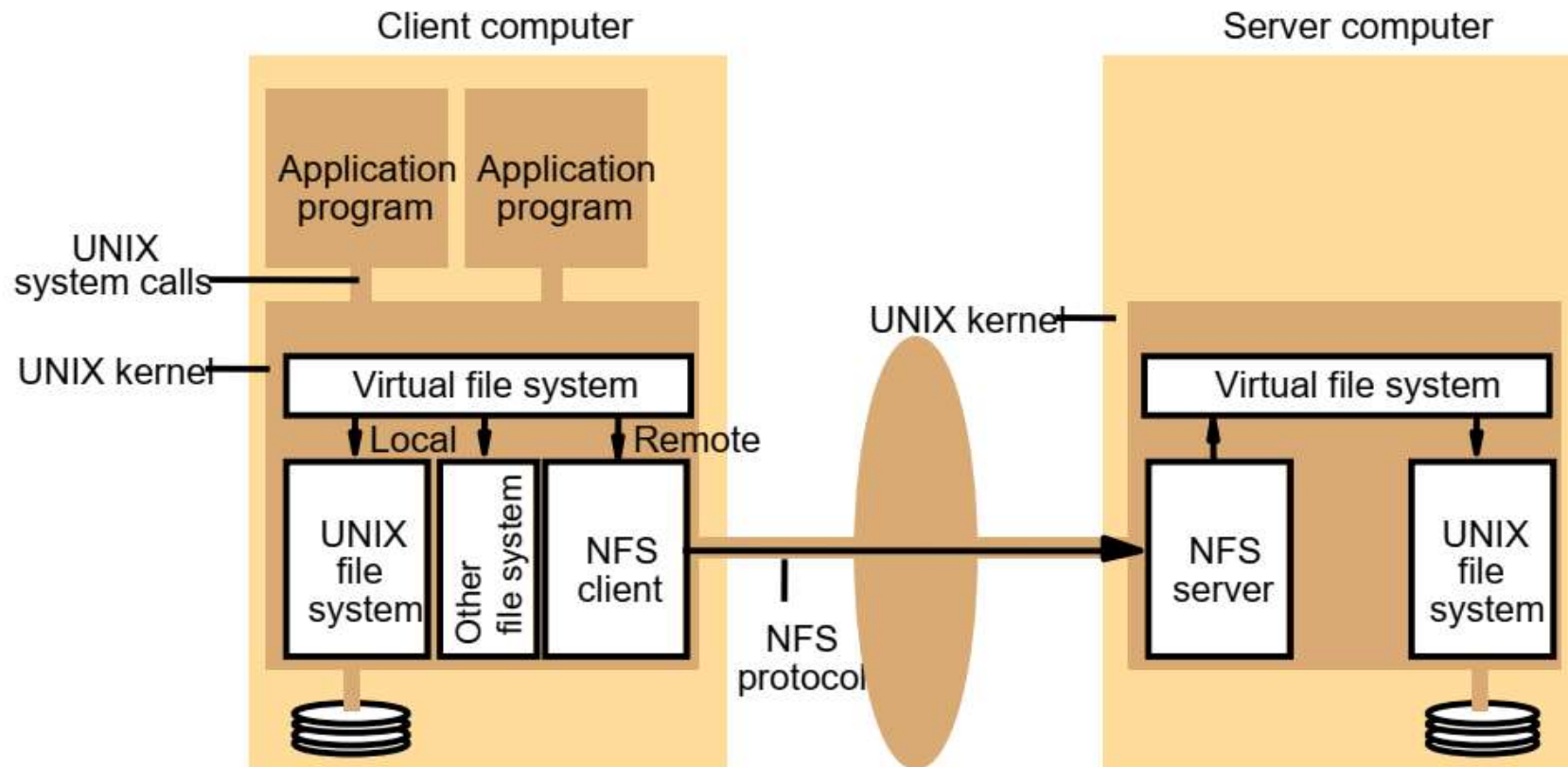
Standar industry untuk jaringan local sejak 1980

Independen terhadap sistem operasi.

RPC dengan TCP atau UDP



# Studi Kasus: Sun Network File System



# Studi Kasus: Sun Network File System

---

Access Control

Client Integration

Pathname Translation

# Studi Kasus: Sun Network File System

---

Access transparency: same system calls for local or remote files

Location transparency: could have a single name space for all files

Scalability: can usually support large loads, add processors, disks, servers...

File replication: read-only replication, no support for replication of files with updates

Fault tolerance: stateless and idempotent

Consistency: not quite one-copy for efficiency

# Studi Kasus: Hadoop Distributed File System

---

Hadoop: framework untuk distributed processing untuk data set yang sangat besar pada computer cluster, menggunakan model pemrograman yang sederhana (MapReduce).

Hadoop Distributed File System (HDFS): distributed file system yang digunakan oleh Hadoop.

HDFS cluster terdiri dari:

- Name Node
- Data Node

# Studi Kasus: Hadoop Distributed File System

---

## Namenode:

- Run on the master node.
- Menyimpan metadata (data about data) seperti file path, jumlah blocks, block Ids. etc.

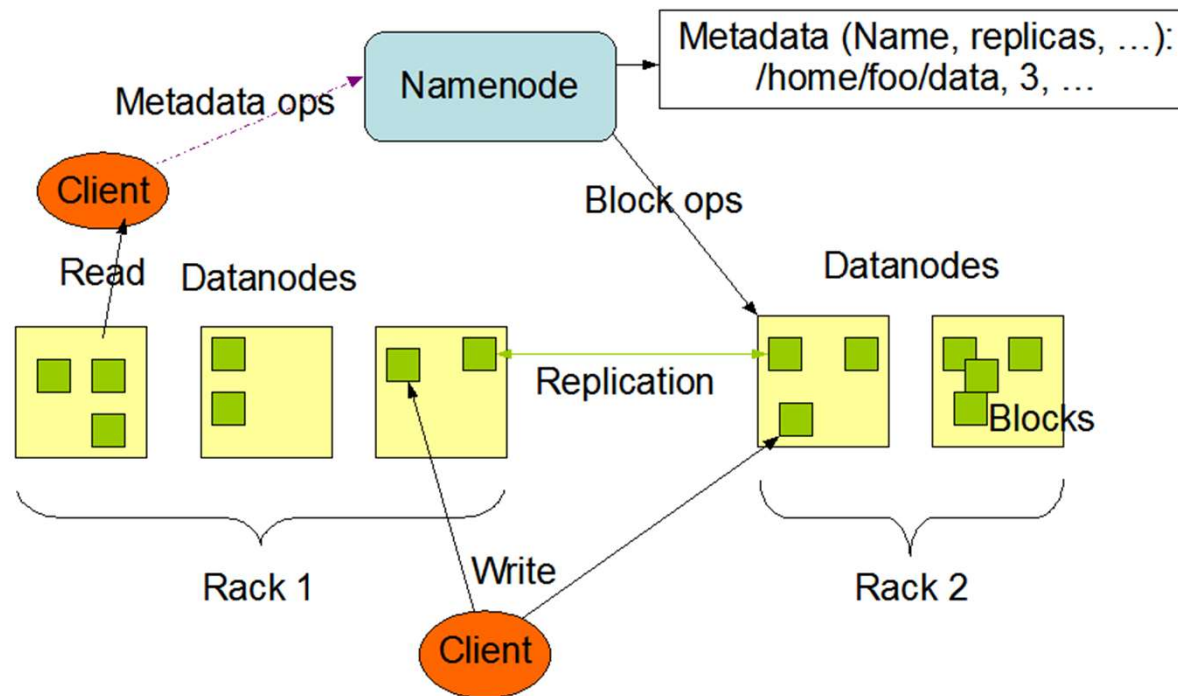
## DataNodes:

- Menyimpan blok-blok data
- Melaporkan kondisinya kepada namenode.

Data yang disimpan pada HDFS memiliki beberapa salinan / replication factor.

# Studi Kasus: Hadoop Distributed File System

HDFS Architecture



# Studi Kasus: Hadoop Distributed File System

---

HDFS didesain untuk file yang sangat besar.

Aplikasi yang kompatibel dengan HDFS adalah aplikasi yang berhubungan dengan data yang besar.

Data ditulis sekali dan dibaca berkali-kali (write once, read many).

Ukuran block pada HDFS adalah 64 MB.

# Studi Kasus: Hadoop Distributed File System

---

Keuntungan menggunakan HDFS:

- Scalable
- Fault tolerant
- Locality of data

Kelemahan menggunakan HDFS:

- Tidak mendukung low latency data access.
- File-file yang kecil menjadi menyebabkan inefisiensi.



# Tugas

---

Dalam sebuah kelompok:

1. Buatlah sebuah single/multi-node Hadoop cluster.
2. Di dalam distribusi Hadoop, terdapat contoh WordCount. Eksekusi wordcount tersebut dengan dataset
3. Laporkan hasilnya dengan susunan:
  1. Langkah-Langkah pembuatan cluster Hadoop.
  2. Penjelasan dataset dan program.
  3. Cara eksekusi WordCount.
  4. Hasil Eksekusi.
  5. Link Video singkat demo
4. Kumpulkan di Kulon (format PDF) sebelum 22 Oktober 2022 pukul 23:59.