

System Co-operatif: معنا تعاون، طبيف يتعاون

Des processus مع بعض

système des fichiers هو ا يعني

client

Request (eg, SSH)

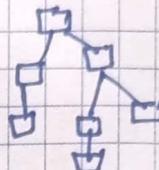
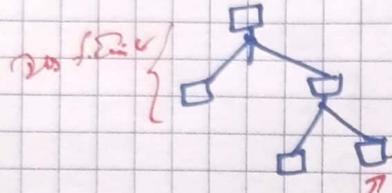
server

point à point

اطفال هنا ذو امتيازات client وrog يتعامل مع Remote Access

نحو ما يعرف بالـ hierarchy .

Un système de fichiers قلنا مللي راسخ



باه اور client يقدر بيا كسيجع لـ server فلزم تتوفر متروط بنزاف هنلا ، فلزم اور client ده يكون عنده ادمعه (droit) في انه بمعدل احنا اور file ، فلزم اور file اين حاب يوصلو يكون فيه اسمو جميعه

قادر ذي حل بسيط ، ذي حل copy اور file

لـ هنلا ملحة ، كوت ذي حل . سجن کلما قادر copy ، النسبة الايجابية تبق بيماراخي .

دخل: ذي حل Montage [نعمزو من اور وندخلو منه اور]
local [ذي حل کانی سنه Ab Strict]
[①. قابلیت دسترسی گامی (Possibilité de l'accès progressif)
②. تغایر فایل (Modification de fichier)
③. قابلیت حذف فایل (Suppression de fichier)]
مشکل آخر: قلاد هنلا يقتو من اور [کات] [گاو] [گاو] و میتو فار [Client] ! سو
رپورتاژ "a"

دخل: ذي حل NFS و تواب تقدرت ذي حل

هذا تسلیل Manipulation

، Modularité et utilisation

NFS يعنی
بسه برجم

نحو NFS
cache client

Processus et Threads:

processus : une instance d'un program qui s'exécute

→ état يقدر بي عادة

→ running

→ Blocked

→ Ready

→ Un Thread d'exécution

→ Un état

→ Des ressources (memoire , fichiers , ...)

يعقد، يكتوفون مع sys co-operatif

ذها تسلیل

Thread: Un processus léger, les threads partagent le même espace mémoire et d'adressage

يسمى IPC
يتم تبادل الموارد بين processes
ويعتمد على التردد
وتحتاج إلى مقدمة نفس

Expre 01

→ Concurrence et Parallelisme
→ Communication inter-Thread facile
→ Optimisation du flux - Coeur.
→ زندر نيمها، تسلسل
→ Threaded

Deux Types de Thread

Thread Système
(niveau moyen)

Thread Utilisateur

يسهل في App تنفيذها
باختصار تفاصيل

AV:

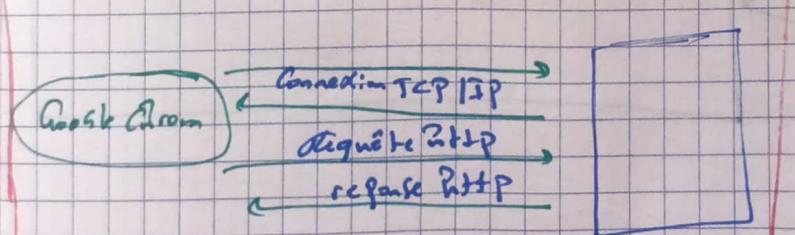
ـ إذا اتى لوك على مثباتها ، فالThread ينتظروها

ـ تسبّب في تأخير

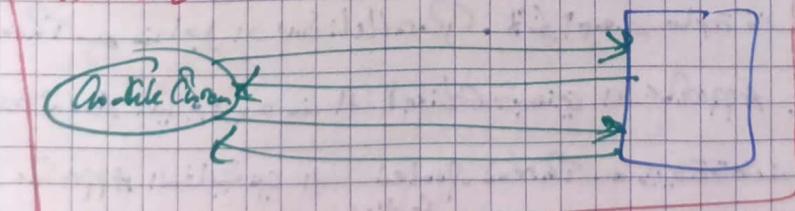
Thread Client-Server لـ performance .

• Multi-threading côté client

Thread 1



Thread 2



: Thread ← Thread

(هذا كلما تتعقب بيكرا (progressive Image) ما يعاني من تقطّعات تطلع بيكرا

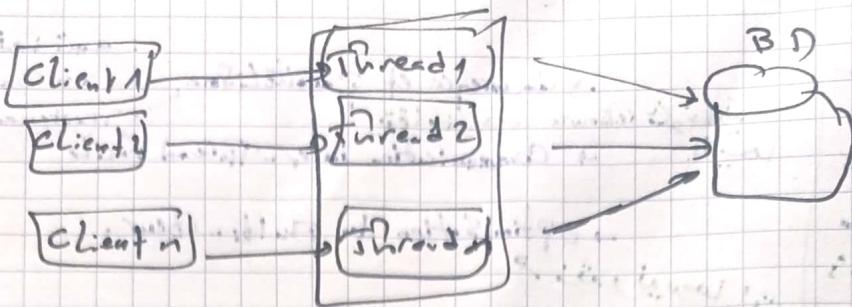
ـ منتظر جهاز غالباً في حالة

AV: . Temps de réponse des serveurs élevé

. Utilisation optimisée des ressources

. Modulaire

مثال على متعدد المدعليات



Thread-Pooled Server ، الخطوة الأولى هي إنشاء Thread Pool

• tasks are concurrent ، الخطوة الثانية هي إنشاء Task Submitters

• tasks are submitted to the Task Queue ، الخطوة الثالثة

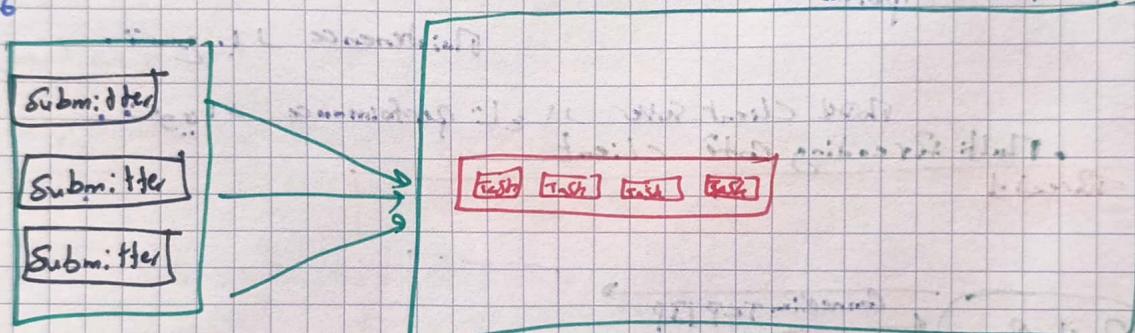
→ Task Submitters

De instances
tasks

→ Executor Service . في هذا Task queue →
• pool Thread file

لكل خدمة قادر على دعم
Debounce
Bottleneck
تحصر عدد threads

Page 16



بما تتيح him Client Requests ، لأن معامله ينجز كل ما في طلب Client .
Server implementation will implement the requests .
وهي من أجل الدليل . Parallelism فار Thread .
أيضاً يجيء بـ Approach . Client ، يعني درجات requests .

Borne et parallélisme في Thread-Pooled Approach in Server

Page 19: qui exécute les tasks FIFO
مقدار task بـ FIFO

Concurrent (متعدد المدعليات)

Fast
less effective
Performance

Limits

- number of threads
- limit on resources

Gestion de la Mémoire Partagée dans un SD.

Mémoire Virtuelle Local et intermédiaire SE, RAN و RAM و Page (en physique) مابينها تغير bind ينطوي على $\text{Cache} \rightarrow \text{Table Pagination}$ Table page

فأولاً نجد SD في Contexte Virtual et Physical nodes يقدرون بـ n مثلاً

(Abstraction cache) Read / Write مكانته.

Large ESPace Mémoire.

Bottleneck مكانته.

Interface Simplifiée.

Mémoire Partagée Distribuée (DSM)

الفكرة كل node يكون خصو بالاتصال local و physique مع باقي nodes و معاً الشريطة $a = b$ فما هي الميزة؟ كل Variable ملحوظة ويسهل التعامل معها

هي نفس ما في Virtual مع فضل كثرة المنشآت

من منظمة هي قبل عادة دفعها بنصف الوقت

استعارات DSM : Page 27

ملاحظة عن جياد

أداة اختبار DSM المفروضة

يكتب كايبر مكتبة

Consistency
Coherence

process 2

```
Begin
  write(a)
  write(b)
End
```

Process 1 : $a = b$

```
Begin
  Read(a,b)
  a = a + b
  c = a - b
  d = c + b
  write(d)
End
```

Node 1

Local

a, b

RP

Node 2

C, D

RP

a, b, c, d

Cache

late

traffic

coherence

consistency

Server يُعطي Copie maître في مستوى الـ N-ary في retard لـ Cache.

(Modifications) كون يكتسب تأخير (client يكتسب تأخير) client يُرسل إلى Server.

Server يتلقي جميع مُطالبات (Modifications) من client.

→ quand mettre à jour la cache server?

(flushing) 3 Variantes pour:

1. Transmission d'un bloc

2. On scanne le cache périodiquement, et on transmet les blocs modifiés.

3. write-on-close

4. Validation du cache:

Est-ce que sa copie locale est consistante avec la copie maître de serveur?

→ Approche client:

لكل Client يتحقق من محتوى Copie المخزنة في Local Cache.

3 approches pour:

1. vérification comme n'accès.

2. accès vérification.

3. vérification continue (الآخر استعلام)

→ approche serveur:

Cache client يتحقق من مستوى الـ MD instance كإذا كان في Server.

(Server)

له ملائمة في التأخير.

Cache fichier → Client يسجل كل الملفات التي كانت في Server.

client, Server تكون Area 1.

Nommage: Examine le caractère utilisé pour référence une entité ou un objet

- une machine
- mode
- serveur
- fichier

Notion de Localisation d'objet

obj. [Object] نظر، نفاذ [Access]

Deux notions lorsqu'il y a nommage:
Abstraction
[Abstract]

Page 44 Exemple de Nommage : Site Web: www.google.com

Adresse : root@mahmed

Nom abstrait : Desktop

URL : https://classroom.google.com/u/1/c/Ntkut...

Localisation = نظر يغطي Site Web

[Domaine] → IP de nœud [عوالي] DNS و هو ملخص

Page 45: قيم [Value] يعبر عن [Name] [nom logique]

plusieurs noms يقدروون [reference] يندر [an object] و يندر [one object]

flexibilité ميزة [flexibility]

Internet: Environnement physique [Physical environment]
stable et localisation physique [Physical location]
→ Localisation et transparence de l'objet
→ Flexibilité des utilisateurs plusieurs noms pour le même objet.
plusieurs objets ont le même nom. (les copies)

les types de noms:

Abstract "Orienté Utilisateur"
جكوبو

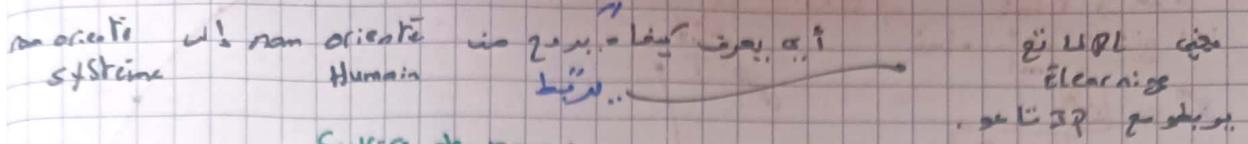
- défini par leur users
- Nom de haut niveau
- pas unique [plusieurs noms pour plusieurs objets]
- nécessite des mécanismes de résolution

Web [Web]
Orienté système"

- nom auto-généré par le sys
- à bas niveau
- ID unique

Qg
pris référence? pr répondre une P

Naming : binding



Service de nom :

les noms (لـ les objets) qui les localisent, les noms يحصل على

compréhensible

de haut niveau

Serveur de nom :

وأوح يكتب فيه [les noms / les IP@]

Page 52 :

à fichier où il aura un serveur de nom pour client

يختلف ويسيرها في localisation

Single Point of failure هو Schema المكلة

ماه نتحقق من backup لـ server of fault tolerance أو
une machine virtuelle

Types de nommage

flat naming

- Non Structured
- Chemin مكتوب
- Aucun info sur sa localisation
- محيط كائنات بلا منطق

Hierarchique . ex. DNS

من انتقاماً بـ hierarchy

Hierarchy

شجرة مطبقة على المكونات

عن طريق الفرع

فيمما متكرر :
Comment répondre une localisation

: 4 Approche

• Broadcasting

• Diffusion

• Forwarding Pointers

• All Joining nodes وال جميع المكونات

→ Forwarding Pointers

Supporte la mobilité des objets

"Trace" لـ les noms

values & keys لـ

important → Table de Hashing distribué et protocol Chord

Un réseau P2P → relatif إلى STS II.

Un réseau Circulaire de même Chord de m-bit

عندما تطلب IP ما في الشبكة

principale successeur (الـ le plus proche)

inConf : non scable

→ Homme-bus

Serveur de nom لـ Homme-Agent Janus.

Un réseau P2P → relatif إلى STS II.

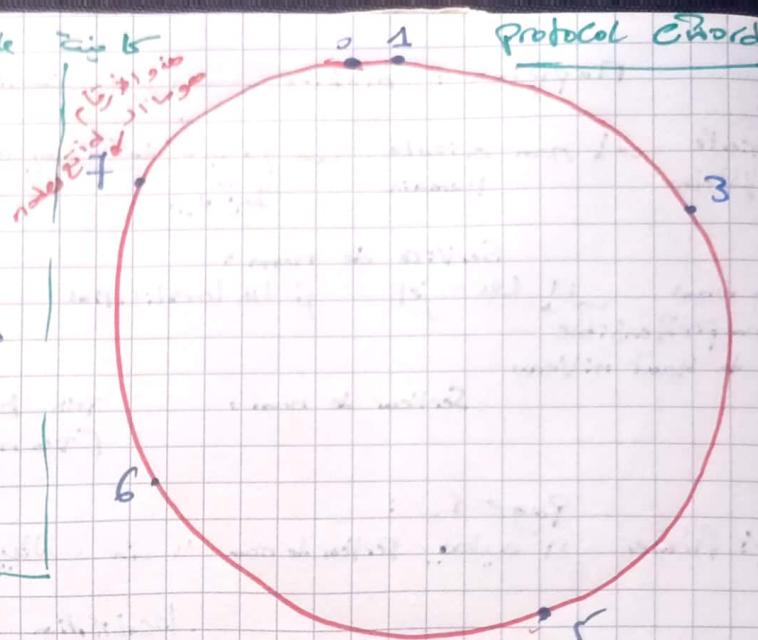
Un réseau Circulaire de même Chord

عندما تطلب IP ما في الشبكة

principale successeur (الـ le plus proche)

كما في الصيغة السابقة key = $\frac{N}{m}$

$$N = 6 \quad m = 3 \quad \text{فإذا جو كوك نحتاج لـ } \lceil \frac{N}{m} \rceil = \lceil \frac{6}{3} \rceil = 2$$



حيث $m=3$ key = 2

نحو $3 \mod 6 = 3$ هو Node 3

أو $i = 3$ المضيفة $i+1 = 4$ المضيفة $i+2 = 5$

$$FT_i = \text{succ}(p + 2^{i-1}) : \text{formula 11}$$

الخطوة هي: $p = 1$
رائج فيه

في الواقع $i = m$

Tableau \rightarrow يوضح عدتنا
للبني فيه
3 lignes

(m lignes) \rightarrow عدد عقدنا هو عدد عقدنا

$p = 1$

1	Succ(1+1)	3
2	Succ(1+2)	3
3	Succ(1+4)	5

$p = 3$

1	Succ(3+1)	5
2	Succ(3+2)	5
3	Succ(3+4)	7

$p = 6$

1	Succ(6+1)	7
2	Succ(6+2)	1
3	Succ(6+4)	3

Complexité:

Approche Linéaire $\Rightarrow N$ nodes

يقترب بعده
 N millions
request

$$\text{Chord} \rightarrow \log_2(N)/2$$

Approche 1 هو الـ Scalable

تقدير توزيع الملفات
تبغيل

i: nbr de lignes

nom \rightarrow les VLs.
localisation

fact & change Consistante: une bonne
Distribution

مثال آخر:

$$I_2 = \{1, 3, 6, 7\}$$

espace de numéros

$$[0 \dots 2^m - 1] \text{ يعبر عن } I_2 \text{ هو } 7 \text{ ، لأن } I_2 \text{ يغطي } 7 \text{ مرات}$$

$0 \dots 7$

$$2^3 = 8 \text{ : قيادة = 3 بـ } b \text{ نحتاج: } (m)$$

$N = 4$ ، فالرسمة أدناه كما يلي :

كل مربع يشير إلى $F_T_p = \text{succ}(p + 2^{j-1})$ fingers print table تابع احتساب امثل

F_{T_3}

1	$\text{succ}(3+1)$	6
2	$\text{succ}(3+2)$	6
3	$\text{succ}(3+4)$	7

دالة succ تكون سندنا obj_P ، رأيناها في fact de Hoare مع اول m وبعطيها قيمة "k"

القيمة هذه هي التي تحدد لي بين متى تتحقق obj_P .

$\text{obj}_P \Rightarrow \text{Hoare}(\text{object Name}, m) = k$ Ex: $k = 6$ \Rightarrow $m = 6$

تطبق $k = 2 \Rightarrow$ succ مرتين succ مرتين succ \Rightarrow $m = 2$.

$$q = F_{T_p}[j] \leq k < F_{T_p}[j+1]$$

Chapitre 02: Communications as un SD

Appel: fibre 光纤

modèle client-server

protocole Application (HTTP) → "Protocol Standard" مع این پروتوكول يتم تبادل معلومات بين client و server

HTTPS : Crypté Sécurisé

اقناع الراهن به تشفير

TLS : Crypté Sécurisé

API: Application Programming Interface API s'appuie sur le protocole HTTP

مجموعة من méthodes pour se connecter à un service avec un logiciel

service

نجل تبادل données

fonctionnalité

REST API: بضم بـ HTTP

les données sont en JSON, donc serveur et client utilisent le même protocole standard

Messages:

Structure: Tête + corps page 19

→ send / receive

4 aspects qui définissent la sémantique de la communication

• Direct / Indirect :

• Bloquant / Non Bloquant

Comm asynchrone

Comm synchrone

• msg buffer: ، (no program non concurrentio - نشيطة ،
destination → message buffer هو عبارة عن (Streaming)

Scheduling

• destinataire → message buffer هو عبارة عن (Streaming)

• Buffer → de la latency / قابلية للتأخير

• fiabilité : تدبر الملاحة No primitive probleme انتقال

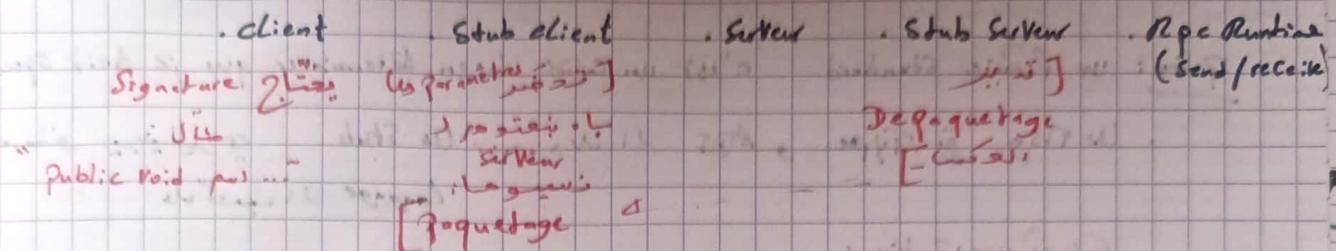
→ Utilisation de Timeout , ACK acknowledgement.

→ Protocoles fiables (TCP,...)

RPC :

→ repos sur le model Client/Server

→ L'Arch. RPC repose sur les éléments :



Page 36 Algo Simplifier de RPC :

1. (Read file) proc ↗ يوجه client.

Distribute

2. يبرهن Client-Stub بـ [Request] فالـ [Request] يبعثه Client.

3. Server ↗ [Request] يبعثه → Rpc Runtime ↗ Client ↗ [Response]

الاتجاه

4. Stub-Server ↗ [Response] ... Server ... -- -- --

5. هنا يرجع [Response] إلى Stub-Server و [Response] إلى Client

6. proc تختلف بـ [Request] و [Response] و [Parameters] (paramètres)

Page 36 : Algo de Fonctionalité.

Exemple d'utilisation de RPC :

→ BD distribuée → IoT

→ Navigation Web → Cloud Computing

Google RPC (gRPC)

5
Google gRPC: Modern open Source High Performance RPC framework that can run in any environment. It can connect services in and across Data Centers

Utilisation gRPC: Netflix . CISCO . Juniper

يستخدم gRPC في Arun Micro-Services

gRPC Reqeust Sw : HTTP/2

Serialisation قديم → Protobuf → JSON معون فهو يدعم

+ متغير في كل تغليف
+ تغليف سلسلة متحركة
+ فئة معينة
+ نوع

format de D
→ A3I

Arch de CORPC page 44

Standardization + Serialization
هو معايير مترابطة
معًا لبيانات
البيانات المترابطة
تحتاج إلى تبادل
بيانات بين المبرمجين

Proto: un fichier standard c'est un code bien déterminé que je doit spécifier

un msg & type , API → nous avons un stub
لغة ما يجب أن تكتب

Synchronization:

problématique dans un SD

- Absence de Dateiation
- pas de vue globale
- pas d'ordonnancement

Pour implementer une SD :

①. SD Centralisé

②. SD Distribué

①: propriété de Consistance \rightarrow لكي
event \rightarrow event : محدثات
 \downarrow
évenement \rightarrow observation \circ $o(e) \rightarrow o(e)$

② \rightarrow Approche non centralisée (Horloge de Lamport)

\rightarrow Temps de Transmission borné . also

also de Lamport : هو ساعة Horloge logique في site كل

. incrementation من event local

. ما يرسل him send pour incrementation (Req) msg

ذاتي لغز

Schéma global page 11

Stockage Distribué de Disques (SAN) et File Disque via une technologie

les espaces Stockage Mutualisation (en base de cloud) Cloud Stockage

Haut disponibilité Globalité Sécurité

page 13 Type de stockage:

Stockage objet

→ Inerte
non structuré
fête données

Stockage bloc

→ SHN
Géantage de la
Technology

Stockage ficher

NAS

Stockage bloc

le système RAID:
taille fixe égale petit

blocks RAID
Striping (Segmentation)

et accès et récupération rapide

→ Haute performance

→ fault Tolerance

→ dédié à SAN (Storage Area Network)

فحائلها

notion:

→ Contrôle Distribué

→ Virtualisation des ressources

→ Très au débit avec le protocole FC (Fiber channel)

→ Fault tolerance (e.g.: RAID)

Arch SAN: Page 17

Les Serveurs → les lecteurs
L'interface logique → des Disques mutualisés
Clients

Le RAID : Technologie de base Stockage Distribué par bloc Utilisant plusieurs Disques
permet la Virtualisation de Stockage

Disques sous forme de blocs (fond)

→ Haute performance : فعالة

→ Fault Tolerance (récupération rapide)

les niveaux de RAID:

- RAID 0 (Striping) : يحقق أقصى تعلم Dual Disques مع تردد مترافقين على الأقل Dual Disques → l'intérieur : espace virtuel باميويه عيّن بأه تقدير خصم يعما.

Stockage par bloc

• fiable : inConv

- RAID 1 (Mirroring) : يحقق D دلالة تغيير في و un seul disque تحكيم خارج دisk الثاني .
l'intérieur : → fiabilité

• RAID 2 راهي الاقصى مع زوج لدار Capacity : inConv

- RAID 5 (Parité) : يتحقق fault tolerance .
• يعتمد على 3 disque مع Calcul de parité qui principle

لديك 3 disque مع واحد دisk داتا .
لديك 3 disque مع واحد دisk داتا .

مثلاً : دisk داتا يخسر .
مثلاً : دisk داتا يخسر .

• دisk داتا

$A_4 = \text{XOR}(A_1, A_2, A_3)$. XOR calculate de parité .

• RAID 10 : striping RAID ويزدوج

زي يزبب في fiabilité

SAN VS NAS

• RAID وfileiers .

الفرق مع SAN . SAN يعني مع fileiers .

• NAS : un dispositif de stockage dédié aux fichiers (Network Attached Storage) يخبي دisk متوفريه دانماش يقدرو الموقعيه بديرو دisk Collabor . Via un réseau .

SNB ; CIFS و NAS يخبي دisk باه يجيبي دisk .

• خاصته : سهل مباشه بيلو . Configuration .

Arch NAS : page 26

• Accès distant : NFS ، لكنه يعتبر أبسط ميكانيزم اللي يخليلون تهير الـ SNB . notion de partage وسنو .

SFS des fichiers

لما يجيء الملفات تكون موزعة على الأجهزة

La SFS - Model de Coherence: VR، وذلك توجهه نحو الـ VR، أي توجهه نحو client

La cohérence تتحقق من خلال نسخة واحدة من الملفات، وهذا يسمى mode cohérence

Contract: Visuellement l'accès concurrent لـ SFS هو متزامن

Un model de cohérence est un contract.

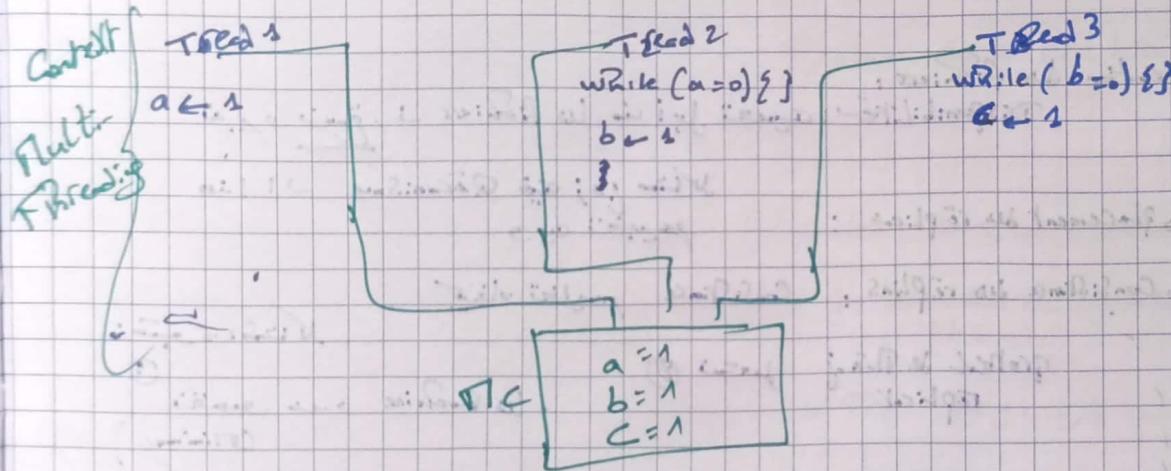
- Coherence Strict:

شدة ال propagation قبل ما تتم القراءة أو الكتابة جميع replicas

نوع Coherence

جميع replicas

نادر و يستعمله ما في



Atomic:
أو متزامن
تحصل كاملة
ومنتهى
خارج

معناها في الحالة معينة لها تكون جميع a=1 مما يعني

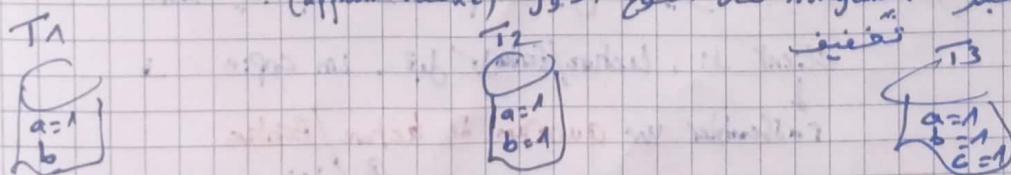
يبيشو فوحا 1 . فالمثال هنا

- Coherence Séquentielle:

هذا دلائل يعنى أن توفر النتائج التي يكون

نفسها على جميع CPU مما يتضمن بنفس الترتيب

(appelé relâché) Allegement du échange البروتوكول



يعين استخدام ذاكرة فردية (Local memory)، لجيم أو كاش، فيه انتظام

بالـ exec sequential

لما يبدوا هما يخدمو القيمة زغ a تكون متساوية 1 ، رايحة انتظام

T لغبي ، ويرمحو يخطو على cach معهم a=1 يعني 1 راجيغ من الـ Bus وتوالي 1

اهم حاجة في هذا الـ Model ، انو يساوي اد écritures كامل ويغافلوا ، les écritures

مابروحت écritures (نسمبودم écritures) les Observateurs Différent . زيرستون . Page 62

[94,95: observers]

Coherence FIFO:

جيمع les écritures كل les écritures هي منتهي الترتيب (الترتيب اليحدث في

écritures

(Execution)

W.R au contraire l'ordre d'écritures . متضمنة .

مثال : Page 63 : تغبيب FIFO . FIFO حبرات PB ، Page 94 تغبيب les écritures . X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 . للتغييرات الترتيب ، كما هنا فاما ما يتم بعد X1 ويمقد X2 بعد X1 .

Coherence Consiste:

متضمنة بحرف Model basé sur les fichiers .

Replication des fichiers:

تغبيب نسخ des fichiers . Disponibilité des fichiers .

هذا الـ mechanism فيه زخم consistency و فيه replication

① placement des replicas :

② consistance des replicas : Consistency

part apprise,

protocole de P.A.-j répliquer

خوب و 壞

③ la machine نفسمو مهام colinage

Approche pessimiste : La cohérence . متضمنة.

optimistic

مهما تعين

▫ primary copy replication: Primary/secondary تفحص

▪ تقدير تعيين المخطىء propagation . قرار غير ما primary second . les exclusives

▪ Voting Ecrite gi Line يطلب ما يعبر قبل ما يطلب des permissions client

client 11 , lecture/écriture . un copie ف

rassemblez un quorum de lecture/écriture R/W

dg: R+W > N; dg.N: no servers

Concessus + Rift (وedo)

diff

Authentification in SD:

Systeme mécanisme qui vérifie l'identité du client et système.

Utilise un mécanisme de cryptographie (SSH par exemple) de cryptologie et une approche à base de clé publique.

(2) Authentification. Sans état de p. sc. (Auth forte) moins de passage possible pour l'accès.

clé publique : pour chiffrer (en chiffrement)

" privée : " déchiffrer (à la clé publique)

الجهاز المعني (الذي عنده)

Server

Page 41: SSH

Windows, Linux : SD utilise Auth SSO protocole : Kerberos

Confiance entre deux serveurs : tickets d'accès et billets d'entrée Server Central (KDC)

Double authentification

Page 43 les phases de kerberos: Page 44

1/ AS als https, Username, client يبعث

2/ يغير فيه الـ client è id AS

Temporary ticket (Ticket Granting Ticket) TGT

- KDC (يهدى هالوارد) une clé

3/ client يروح بحد اد (TGT, clé) ويعتني

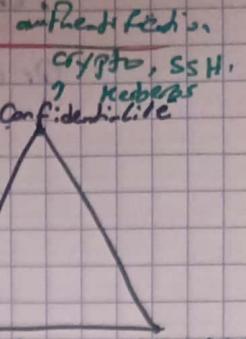
TGS

4/ Client يبعث id AS

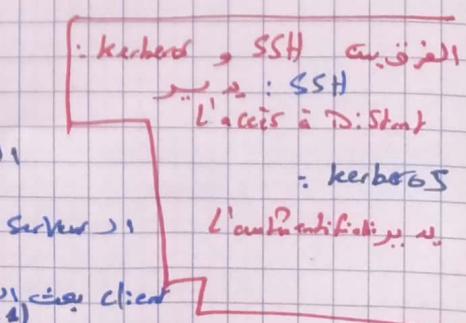
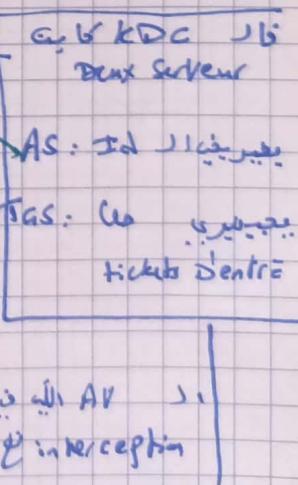
5/ Client يبعث id AS

6/ Client يبعث id AS

7/ Client يبعث id AS



Integrity	Availability
Auth	Replication
Concessus	fault tolerance
Modèle de	load balancing
Coherence	



Chiffrement:

Pré-crypter un message:

• Algo Symetric (une seule clé)

→ DES 56 bits de clé

→ Triple DES

→ AES 128 bits de clé

• Algo ASymetric (2 clés)

→ RSA , clé de 1024, 2048, ...

• Hashage

→ MD5 → SHA-1 , SHA-2

Page 47

RSA:

SSL مستعمل بكثرة فار

TLS : (Transport Layer Security)

عبارة عن بروتوكول يضم بروتوكول HTTPS و SSL و ينتمي إلى عائلة web

• Chiffrement Sym / Asym ()

4 phases de TLS :

1/ TGS Round-trip

2/ Certificate check (clé publique du serveur)
جامعة Algo و انتيكو Version → clé publique

3/ L'échange des clés

4/ .. de données (Algo sym)