



Rapport de Similitude Turnitin

Introduction aux BDs NoSQL par Khaled Jouini

De Quick Submit (Quick Submit)

Traité le 13-déc.-2024 11:59 PM CET

Numéro : 2551742818

Nombre de mots : 7981

Indice de Similitude	Similarité par source
0%	Internet Sources: 0% Publications : 0% Copies de l'étudiant : 0%

Sources :

Aucune source ne correspond à ce rapport.

Texte de la copie :

n(goa(rérc)es.{ap_ leaiuidsr,saut.ntlerhueéinrasg,li.tsaahacc;tettuoerr+usr+.)sl)e{nr{getsh}=;}res +
acteurs[j];} h a d @ Exercices Exercices Écrire en pseudo-code les fonctions Map/Reduce permettant de : c
o m 1 Afficher les titres des films du genre "drama" par année il. 2 Trouver pour chaque acteur la liste des
acteurs avec lesquels il a tourné des films 4j.ATrfokiuvcvheerrfelesgfrialmpshepladeregseconore-aertistpeasr
année g m a 3 h a d @ Chapitre 4 - Réplication Réplication : pourquoi et comment? Cohérence forte vs.
Eventual Consistency o m Théorème CAP Heartbeats, Failover et élections c La réplication dans la pratique
- Cas de MongoDB Quiz m a il. h a le d @ g j.k Réplication : pourquoi et comment? Réplication : dupliquer
les données sur plusieurs serveurs pour assurer • Haute Disponibilité : Capacité d'un système à rester
fonctionnel et accessible • mToélémaenecne caausxdpeapnanennse:
Cdaepcaerctiatéindseserrévceuprsé.rer les données, perdues. o m • Répartition de la charge : Améliorer
les performances en répartissant les accès aux données entre plusieurs serveurs. m a c
Da**njs.EFLglaeséesksqesptéunluocetpinreniatltculeerotrsemnd.ssaepsnssolessnxylteeslgtsédécemenluéesessrataec:lréecsmcédriesittcnuroitr
h a d @ g d'écriture, verrouillage, etc.). Réplication : pourquoi et comment? Réplication Master/Slaves
(Leader-Based Replication) • Lorsque les écritures se font sur un seul serveur, ce serveur est appelé
Master. c o m • Les autres serveurs, appelés Slaves, contiennent des copies et répondent à éventuellement
aux requêtes de lecture.
**j.PIPeaasknnénceeritsduurreuMnsa\$sltaevrle:eu:nleSslaysvteèmpueuctoèntretnuperoàmfunpctoiurnlenermetpàlacaessru(réelrelects
et h a d @ g m Mécanisme de Synchronisation : Fichier Journal (Log) Léecristuysretèsm.es de réplication
utilisent des fichiers journaux (log)iplou.rconsigner les a c o m Chaque modification est enregistrée dans un
fichier journal et les répliques rejouent les modifications à partir de ce log.
Llledaséjéaojclue.rrimntukaerlenisstatauionlénaslaptoveierremsrée.lcteupeèssree rleamloegnderu
àmdaestseér,csriintuornesds'éuqnuselanvtieelles plus rapides que @ g m h a d Chapitre 4 - Réplication
Réplication : pourquoi et comment? Cohérence forte vs. Eventual Consistency o m Théorème CAP
Heartbeats, Failover et élections c La réplication dans la pratique - Cas de MongoDB Quiz m a il. h a le d @
g j.k Réplication synchrone et cohérence forte Réplication Synchrone • Une écriture n'est validée que si elle
réussit sur toutes les répliques, garantissant • aCion*shiqTroueuentcetoelufctoeturstreelersencvoopieielas
dvaelseudrodnlenéadesroninètretoéucjroituurres vidaelidniéltieq..ues. • Entraîne un temps de latence
élevé. m a c o m • Mode adopté par les bases relationnelles (E.g. Oracle Data Guard) Difficile à assurer
dans les environnements distribués à grande échelle, où les pannes sont fréquentes et les répliques ne
sont pas toujours atteignables. h a d @ g Réplication asynchrone et cohérence à terme (Eventual
Consistency) Réplication Asynchrone : Les écritures sont confirmées sur le maître sans attendre (tous) les
slaves.
Co*héLgreansracdneotniàentédeeermsdseeérl(aoEin.vteéntvueanltuCeollnesmisetenntccyo)hérentes sur
toutesilèls.répliques, mais sans c o m ⇒ Latence réduite, performances accrues, tolérance aux pannes;
mais Incohérence temporaire possible. d @ g m a j.k le h a Réplication synchrone vs. Réplication
asynchrone Critère Synchrone Asynchrone Cohérence Forte À terme Latence Élevée Faible Tolérance aux
pannes Faible Élevée Mécanisme de réplication CScojam.lapbkleilixité de
mRiéseplelicneacetiounvrseynchrone vs. PRIÉuÉspldivcifaefiectioilen asynchrone d @ Blocage jusqu'à
confirmation g m Réplication différée Plus simple a Plus facile c o m a il. h Théorème CAP Réplication :
pourquoi et comment? Cohérence forte vs. Eventual Consistency m Théorème CAP Heartbeats, Failover et
élections c o La réplication dans la pratique - Cas de MongoDB le g m a il. d @ j.k h a Théorème CAP Le
Théorème CAP • Le théorème CAP indique qu'un système distribué ne peut pas garantir simultanément : **
CDioshpéorneibncilieté((CAovnasiltsatbenilicty)) ::
CTohuatqeuselerserqépèliqtueerescvooiiteunntelersémipoèlnm.seesmdèomneéens. cas de c o m
panne. m a • Tolérance au Partitionnement (Partition Tolerance) : Le système continue de fonctionner même
si la communication entre les serveurs est coupée.
**j.LeLeetksstosRylDéstrBèaMmncSeepsrNaivuolèSpQlgailLeertndittioilainvencenotmhcéehrneotni,siacreudedeutéxtarpimdroeisppnrtioédnti
h a d @ g Théorème CAP Théorème CAP m a il. c o j.k le h a d @ g m Réplication : pourquoi et comment?
Cohérence forte vs. Eventual Consistency m Théorème CAP Heartbeats, Failover et élections c o La
réplication dans la pratique - Cas de MongoDB le g m a il. d @ j.k h a Tolérance aux Pannes, Heartbeats et
Failover Les serveurs échangent des messages de heartbeat pour vérifier leur état.
SSdiéuucnlessnleacrvvhééutopmmoubareitpreonnmepoarénvnpnoeirn,udrinepnslluansveaesuecxophmaesmasreetbneoutsv,euanur
c o m de consensus, dont notamment PAXOS et plus récemment RAFT. De préférence nombre de votants
impair (pourquoi?)
Cri**tjèr.LLlee'aséuktprdarreôtiolsderéitééolesudcynecticosahnpsearolcvneiietsuéar)ts.io(ncedretasinréspsleiqrvueeusr(slaperéupvleiqnuteétl
d @ g m a h a Algorithme de Consensus RAFT - Élection du leader c o m m a il. Figure reprise de [OO14]
Dé**tjèc.CreLdetetehçokdvafnooeqilndilutopiereêwaslteefspoeamdalsroeenewunmhtveeraeprsaanlldarsuticfebsopéleero.maastdeepeda
h a d @ g • Un candidat lance une élection en augmentant son terme et en envoyant des requêtes de vote
(RequestVote) à tous les nœuds. Algorithme de Consensus RAFT - Élection du leader Vote d'un nœud : Un
nœud recevant une requête de vote **