Rapport de Similitude Turnitin

Streaming (Big) Data, Lake House et Architecture en médaillon par Khaled Jouini

De Quick Submit (Quick Submit)

Traité le 13-déc.-2024 1:19 PM CET
Numéro : 2551211934
Nombre de mots : 8178

Indice de Similitude	Similarité par source	
1%	Internet Sources:	1%
	Publications :	0%
	Copies de l'étudiant :	0%

Sources :

1 1% match (Internet depuis le 02-déc.-2020)
http://www.wikiwai.com/2019/11/12/elaboration-dun-data-lake/

2 1% match (Internet depuis le 12-oct.-2021)
http://ailab.ssu.ac.kr/rb/?a=download&m=upload&r=home&uid=1604

Texte de la copie :

Streaming (Big) Data, Lake House et Architecture - Étude de cas avec Apache SepnarkmStLréuackdteuar-edillSotrneaming, Aipla.che Kafka et Delta a c o m Khaled Jouini j.k lej.klShITCaolme,-Ud-n--@iv-e-g-rs-imt-y-a-ofiSolu.ssecom @ g m h a d m a il. c o j.k le h a d @ g m Plan du cours (1/2) 1. Stream Processing et Streaming Data - Étude de cas avec Apache Spark 1.1. Stream Processing, quésaco? 1.2. Structured Streaming - Concepts de base 1.3. Structured Streaming - Concepts avancés 1.4. Structured Streaming - Agrégation temporelle 1.5. Machine Learning et flux de données c o m 1.6. Session illustrative il. Quiz m a 2. Streaming Pipeline - Étude de cas avec Kafka g j222222Q.....123456u.....ikzACPMASeaploogsramstdoiictoerpiiohtnsohnedsmInlaleKueenasmlttifdvrskaelreadatnei,eivCtsqeoeKountanérsfs astRioAnFTdes topics @ h a d Plan du cours (2/2) 3. Architecture Lambda 3.1. Principes

13.2. Limitations de l'Architecture Lambda 3.3. Transition vers l'Architecture

id9aeoe5fB/ltk.-paa9ad.bfLNusa/okPdev.led(mhg_tbdteartpa2sb0:r2i/4c./ksd.epdlft.a.io/), h a d @ Section 1 - 1. Stream Processing et Streaming Data - Étude de cas avec Apache Spark 1. St1r.e1a.mStrPeraomcePsrsoincgesestinSgtr,eqauméisnagcDo?ata - Étude de cas aivle.cApache Spark c o m 1.2. Structured Streaming - Concepts de base 1.3. Structured Streaming - Concepts avancés 1.4. Structured Streaming - Agrégation temporelle d @ g m a j11Q...56u..ikzMSeasscIohnineilluLsetraalrtneivineget flux de données h a 1.1. Stream Processing vs. Batch Processing Stream Processing : méthode de traitement des données en temps réel, où lCesomdopnanréaeisosnsoanvtetrcailteéetrsaaiteumfuernettpàamrloetssu:re qu'elles arrivieln.t,plutôt qu'en lots. a c o m ▶ Batch Processing : Les données sont collectées sur une période, puis traitées en une seule fois. ▶ j.Srséotkrcneeatprmatiiotpenrmo,rceéendstu.sliisneagn:tLaeinssdiloanlnaéteenscsoenetntrtareitél'eapspenarcitioonntidneu ld'éèvsélenuerment et h a d @ g m 1.1. Stream Processing vs. Batch Processing ▶ Réactivité en temps réel : Permet de prendre des décisions immédiates (ex : détection de fraude). c o m ▶ Traitement continu des données : Traite les données non structurées ou semi-structurées à mesure qu'elles arrivent (ex : flux de capteurs, logs). g m a il. ▶ j.Apadkraspetcéoanudxegarvalenecdeusnevoslcuamlaébtriileités :hSouripzopnotrateled.es millions d'événements @ h a d 1.1. Comparaison : Data at Rest vs Data in Motion Caractéristiques Data at Rest Data in Motion m Données Statiques Dynamiques, en temps réel o c Stockage Bases de données, en- trepôts de données, Data EoCnuondtrtaaiinnnluss.ioutnusupernipleteeslimnreépsseraéuexl Lakes m a Traitement Par lots (batch process- ing) g Latence Haute (analyse différée) lVfAeixonelaulmysee dimeploogrtsa,nratppmoartiss d @ Faible (analyse immédiate) a Volume Volume potentiellement infini Exemples h Transactions financières, CTejac.shdkn'oultoilgisiaetsion financiers loT, réseaux sociaux Archivage, Data Ware- Détection de fraudes, housing monitoring temps réel HDFS, bases de données, Kafka, Spark Streaming, Data Lakes Flink Table: Comparaison enKthraeleDdaJtoauainit Rest et Data in Motion 1.1. Cas d'utilisation du Stream Processing ▶ Surveillance des systèmes loT : Analyse des données de capteurs en a c o m ▶ tSeymstpèmsreéselfipnoaunrcdieérste:cStuerrvdeeillsanacneomdaeslietrsa.nsactions iplo.ur détecter la fraude instantanément. ▶ ▶j.AiPcnuanckbmaidllipyecsanitetégsdnedeenesslspligoéugcnlbueselircei:tiénAtanteieranemlsyd.psiersedrcéete.slc:lDicésteencttieomnpdser des h a d @ g m 1.1. Défis du Stream Processing ▶ Faible Latence : Traitement des données presque instantané pour offrir a c o m ▶ uTonleérraénaccetivaituéxépleavnénee.s : Assurer que les données neilso.nt pas perdues ou doublées en cas de panne du système. ▶ ▶j.CdGto'okeuénstctsehiiosnetnacadnoscesuuserddadeoneltnspelnaedéropteunrsérnbtcéaaiesrtidiosoinv:neGd(seeax:sraalnrénctésttiulrgyld reaoertnssl.ceréessdupolrtoancntseéesexssianacgrtr)si.vmanêtmeen erentacrads h a d @ g m 1.2. Apache Spark Structured Streaming ▶ Structured Streaming ▶ Framework de traitement de flux de données, scalable et tolérant aux ▶ pAPuaotvilsnaissnniébetsailsitpgécoeodusner:slteefrauxtiiprrtearseiultasersmlijeooeinmnntdoutetbreesauettrrcnadhtitr'.eeemxdéeocnnunttsié c o m ▶ ▶ Traitement batch avec Spark : 1 2 2 irirrnSeSetpts▶srrDjeuuetFaa.IDTtmm.rDaFkwDDFitr=FFei=t==mespi.esinSpafntpoarrturekrmeatk.Dnra.meFrtseaD(l.tda"rFseepde..aafSlsoe ((,a"""".rd"jl,koses"ai:ssodgitn-n"g("pan)sala.ot"llouh")ar").dcw).ew(h-"ehpseraeortue("hr("s"cise)ig-npgaanltahl>">)151"5")) @ g m a 3 h a d 1 3 r S t r e a m D F . w r i t e S t r e a m . f o r m a t ("parquet") . s t a r t ("dest-path") ISITCom Khaled Jouini 1.2. Modèle de données m a il. c o j.k le h a d @ g m 1.2. Modèle de données Concept clé : Le modèle de Spark Structured Streaming repose sur une abstraction