Big data Exam

Ερώτηση	Απάντηση/Εξήγηση
Διαφορές GFS - HDFS	HDFS: GFS: Open Source Developed by Google NameNode - Datanode MasterNode - Chunk Server 128MB default node size 64MB default node size Μόνο Append Random File write γίνεται Write Once - Read Many Multiple Write - Multiple Read
HDFS πως εξασφαλίζεται Fault Tolerance;	– Ένα αντίγραφο στον τοπικό κόμβο. – Δεύτερο αντίγραφο στο ίδιο rack – Τρίτο αντίγραφο σε απομακρυσμένο rack – Επιπλέον αντίγραφα τοποθετούνται σε τυχαίους κόμβους
Πώς έχει υψηλή διαθεσιμότητα το HDFS;	- Κρατάει αντίγραφα κάθε block από κάθε αρχείο σε 3 διαφορετικά nodes (by default). - Γρήγορη ανάρρωση από σφάλματα - Αντιγραφή της κατάστασης του Master
Bloom Filters	* Χρησιμοποιείται για να καθοριστεί αν κάποιο SSTable περιέχει δεδομένα από συγκεκριμένο row χωρίς να το ανακτήσουμε. * Πιθανά τα false positives, ποτέ false negatives.
Columnar vs Row databases (query use cases)	Row-Oriented DB: Συνήθως είναι optimized και είναι καλές για OLTP (online transaction processing) δηλαδή γρήγορες αλλαγές π.χ. INSERT, UPDATE, DELETE Column-Oriented DB: είναι καλές για OLAP, πιο σύνθετα aggregations π.χ. SELECT A ή SELECT AVG(A)
Lazy Execution (Τι είναι αυτό και ποια τα οφέλη)	* Γράψιμο σε υψηλότερο επίπεδο των οδηγιών μετασχηματισμού, οι οποίες εκτελούνται on-demand (με το collect η κάποιο άλλο τρόπο). * Δημιουργεί ευκαιρίες βελτιστοποίησης.

Ερώτηση	Απάντηση/Εξήγηση
Lazy Transformations	Ότι και το πάνω, οι μετασχηματισμοί σε ένα RDD είναι lazy (ενδιάμεσες λειτουργίες), γιατί περιμένει να δώσεις και άλλους μετασχηματισμούς, για να δει αν μπορεί κάτι να το αποφύγει ή να το τρέξει παράλληλα. Η εκτέλεση γίνεται με την τερματική λειτουργία (action).
Map Reduce κοντά στα δεδομένα (Data Locality)	Θέλουμε οι υπολογισμοί να γίνονται κοντά στα δεδομένα (ίδιο κόμβο π.χ.) ώστε να αποφεύγεται η μεταφορά μεγάλων δεδομένων εκεί που γίνεται ο υπολογισμός. Έτσι μικραίνει η συμφόρηση του δικτύου και αυξάνεται το throughput του συστήματος. Γίνονται spawn λιγότεροι executors.
Map Reduce τι είναι η Partition συνάρτηση;	Βρίσκεται μετά το map και πριν το reduce, διαμοιράζει τα key-value pairs, ουσιαστικά hashάρει (hash mod) τα κλειδιά και σπάει το εύρος τιμών των κλειδιών για παράλληλες εκτελέσεις reduce. Εκτελείται περιοδικά.
Map Reduce τι είναι οι combiners;	Βρίσκονται μετά το map (πριν την partition) και λειτουργούν σαν ένα mini-reduce τοπικά στη μνήμη, σαν βελτιστοποίηση για να μειώσουν την κίνηση στο δίκτυο. Ουσιαστικά, συνοψίζουν τα outputs με ίδιο key.
Map Side - Reduce Side Join τι είναι πιο γρήγορο	* Το Map-Side είναι πιο γρήγορο αλλά προϋποθέτει ταξινομημένα datasets ως προς το κλειδί συνένωσης. * Το Reduce-Side είναι πιο αργό αλλά είναι γενικού σκοπού.
Map Side Join (== Sort Merge Join)	Λειτουργεί εάν δύο σύνολα δεδομένων είναι συν-διαμερισμένα και ταξινομημένα με το κλειδί συνένωσης. Map πάνω από το ένα σύνολο δεδομένων, ανάγνωση από άλλο αντίστοιχο διαμέρισμα Δεν απαιτούνται reducers (εκτός αν θέλουμε να κάνουμε κάτι άλλο)
Reduce Side Join	> Map και στα δύο σύνολα δεδομένων > Εκπομπή πλειάδας ως τιμή με κλειδί συνένωσης ως ενδιάμεσο κλειδί > Το πλαίσιο εκτέλεσης συγκεντρώνει τις πλειάδες που μοιράζονται το ίδιο κλειδί > Εκτέλεση της συνένωσης στην πλευρά του reduce Παράλληλο sort-merge (ταξινόμηση-συγχώνευση) στις παραδοσιακές Βάσεις Δεδομένων (όχι ταξινομημένα, όχι broadcast, όχι cache)

Aa Ερώτηση	Απάντηση/Εξήγηση
Hash Join (==Broadcast Hash Join)	Φόρτωση ενός συνόλου δεδομένων στη μνήμη σε ένα πίνακα κατακερματισμού (hashmap), με κλειδί το κλειδί συνένωσης Ανάγνωση του άλλου συνόλου δεδομένων, διερεύνηση (probe) για κλειδί συνένωσης. Λειτουργεί εάν R << S και το R χωράει στην RAM. **Σαν διαδικασία MapReduce :** Διανομή του R σε όλους τους κόμβους (πχ, μέσω της DistributedCache) Map στο S, κάθε mapper φορτώνει το R στη μνήμη και δημιουργεί το hashmap Για κάθε πλειάδα στο S, έλεγχος (probe) του κλειδιού συνένωσης στο R Δεν απαιτούνται reducers (εκτός αν θέλουμε να κάνουμε κάτι άλλο)
MapReduce on Very Large vs. Very Small Dataset	Το MapReduce προορίζεται για batch processing, δεν αξίζει σε μικρά επειδή προσθέτει overheads. Δηλ. ότι κερδίζουμε σε χρόνο λόγω των υπολογισμών το χάνουμε στην επικοινωνία μεταξύ των κόμβων.
MapReduce Shuffle & Sort	**Shuffle**: Στέλνει όλα τα key-value records με το ίδιο key στον reducer που του υποδεικνύει ο partitioner, **Sort**: ταξινομεί όλα τα key-values σε έναν reducer βάσει του key. (Αν έχει οριστεί συνάρτηση combine τρέχει πριν το shuffle.)
Narrow-Wide Dependencies (RDD)	**Narrow :**Το κάθε τμήμα του γονικού RDD χρησιμοποιείται από το πολύ ένα τμήμα του απόγονου RDD (συμμετέχουν λίγοι workers, απλό) π.χ. `map, filter` **Wide:** Το κάθε τμήμα του γονικού RDD χρησιμοποιείται από περισσότερα από ένα τμήματα του απόγονου RDD (αργό, θέλει shuffle and sort) π.χ. `groupByKey`
Query Optimization (SQL)	– Εύρεση ενός πλάνου υπολογισμού – Αναζήτηση βέλτιστου πλάνου (μεταξύ πολλών πιθανών)
Για εγγραφές σταθερού μήκους, πως κάνουμε προσπέλαση (σε βάση)	Η αποθηκευμένη εγγραφή i ξεκινά από το byte n * (i - 1), όπου n είναι το μέγεθος κάθε εγγραφής.
Περιγραφή χειρισμού ενός SQL query (πλάνο εκτέλεσης)	Query → Parser → Translator → Σχεση → Λογικο πλανο → Optimize (Join, Select) → Διαλέγει → Φυσικο πλανο
Τι συμβάλει στο κόστος ενός SQL ερωτήματος;	I/O, μεταφορές απο δισκό στην μνήμη, CPU, Δίκτυο επικοινωνίας

Ερώτηση	Απάντηση/Εξήγηση
Τι συμβαίνει όταν αποτυγχάνει ενας Mapper ενώ στέλνει τα δεδομένα σε έναν Reducer και κόβεται στη μέση η όλη διαδικασία αποστολής;	Επανεκτελείται αν δεν υπάρχουν αντίγραφα, αν υπάρχουν, speculative execution (όποιος προλάβει να τελειώσει πρώτος)
Τι join σε distributed σύστημα αν έχεις ένα μεγάλο κι ένα μικρό dataset;	BroadcastHashJoin (== hash Join) εφόσον το μικρό χωράει στην μνήμη.
Τοπικά, τι join αν είναι ταξινομημένα τα 2 σύνολα;	Εφόσον είναι ταξινομημένα, map-side join το οποίο λειτουργεί υπό αυτή την υπόθεση.
Spark Hash Shuffle vs Sort Shuffle	Το Hash Shuffle δημιουργεί 1 αρχείο για κάθε reducer, ενώ με το sort shuffle εξάγεται ένα μόνο αρχείο που έχει δεδομένα ταξινομημένα ανά "reducer" id και είναι ευρετηριασμένο. Για μικρό αριθμό reducers το hash shuffle είναι πιο γρήγορο, γι'αυτό χρησιμοποιείται ως κύριο το sort shuffle και σαν εφεδρικό το hash shuffle όταν ο αριθμός reducers είναι μικρότερος από ένα κατώφλι (default 200). Αν στο sort shuffle δεν έχω αρκετή μνήμη να χωρέσω όλο το map output τότε spill δεδομένα στο δίσκο.