

Course Export - Big Data Systems HPI

University

Uni-Potsdam

Semester

None

Created By

Rodlik

Description

In this course, we want to provide an overview of big data technology. We will discuss the big data software stack, which forms the basis for most big data systems and then give an overview of the variety of big data processing systems. You will learn the composition of big data systems as well as the inner architecture of each system.

<https://hpi.de/rabl/teaching/winter-term-2020-21/big-data-systems.html>

Quiz 1 Map Reduce

Why is MapReduce so popular?


MapReduce significantly eases the development of parallel programs.

Which of the following statements about classical MapReduce is true? 1. MapReduce takes care of skewed keys. 2. MapReduce starts redundant workers for better performance. 3. MapReduce users need to handle node failures. 4. MapReduce directly streams data from the Mapper to the Reducer.

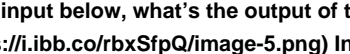

1. MapReduce starts redundant workers for better performance.

What is the output of following mysterious map reduce algorithm? !

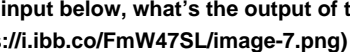

Tuples containing the number of occurrences of words per word

What does the following MapReduce job do? Select the correct answer. You are given a document corpus with *n* documents and *m* unique words. The order of the MapReduce job is: MAP1, REDUCE1, MAP2, REDUCE2. !

It calculates the average word occurrence across all documents, i.e., one result per document corpus.

Given the input below, what's the output of the following mysterious map reduce application? ! Input: 

(192.168.1.13, 4)

Given the input below, what's the output of the following mysterious map reduce application? ! Input: 

(192.168.1.5, <192.168.1.1, 3>); (192.168.1.42, <192.168.1.13, 4>)

You want to sort a 100 GB (1GB = 10⁹ bytes) file that consists of 50 000 blocks using Two Phase Multiway Merge Sort. Your main memory can fit 3200 blocks. * How many I/O-operations does it take to sort this file? * What is the exact maximum file size that can be sorted? Provide your answer with two decimal places.

- **200000** --> We have 50 000 block reads and 50 000 block writes for each phase. So $4 * 50\,000 = 200\,000$ I/O-Operations.
- **20.47TB** --> The maximum file size is determined by $\#blocks\text{-in-memory} * (\#block\text{-in-memory} - 1)$. This means that in phase 2 we can process 3199 sorted partitions $* 3200$ blocks = 10236800 blocks. One block has a size of 2MB (100 GB / 50'000 blocks). If we multiply this with the number of blocks that we can process, we have a 2MB $* 10236800$ blocks = 20.47 TB file size limit.


Quiz 0 Distributed Systems

Does the following equation hold, i.e. are the operators “selection” and “projection” commutative in this case? R is an arbitrary relation with possibly more columns than c1, c2, c3. $\pi(c1, c2, c3) (\sigma(c2=\text{literal}) (R)) = \sigma(c2=\text{literal}) (\pi(c1, c2, c3)(R))$ Select one: * Yes, always. * Conditionally, if we can ascertain that there are no duplicates in the (c1, c2, c3) of R. * No, never. * Conditionally, if the column c2 has only distinct values.

Yes, always

Consider a system containing 4 identical disks. Match the different performance characteristics ('Read + Write Speed') compared to a single disk to the specified RAID setup: * RAID 1+0, where disks are pair-wise mirrored * RAID 5, across all disks * RAID 0 across all disks

- RAID 1+0, where disks are pair-wise mirrored → 4x read performance. 2x write performance.
- RAID 5, across all disks → 4x read performance. 1x write performance.
- RAID 0 across all disks → 4x read performance. 4x write performance.

Consider the following two relations. R contains students and the lectures they attend. S is a set of lectures.  What does the following SQL statement do? `sql SELECT r.Name FROM R r WHERE NOT EXISTS ((SELECT s.Lecture FROM S s) EXCEPT (SELECT DISTINCT v.Lecture FROM R v WHERE v.Name = r.Name))`

Return all Students that attend all lectures in S. (Result = Julia, Amanda)

Evaluate the query from the inner to the outer part.

```
( SELECT DISTINCT v.Lecture FROM R v WHERE v.Name = r.Name )
```

The lectures the student attended.

```
( SELECT s.Lecture FROM S s )  
EXCEPT
```

Minus all lectures


```
SELECT r.Name  
FROM R r  
WHERE NOT EXISTS (
```

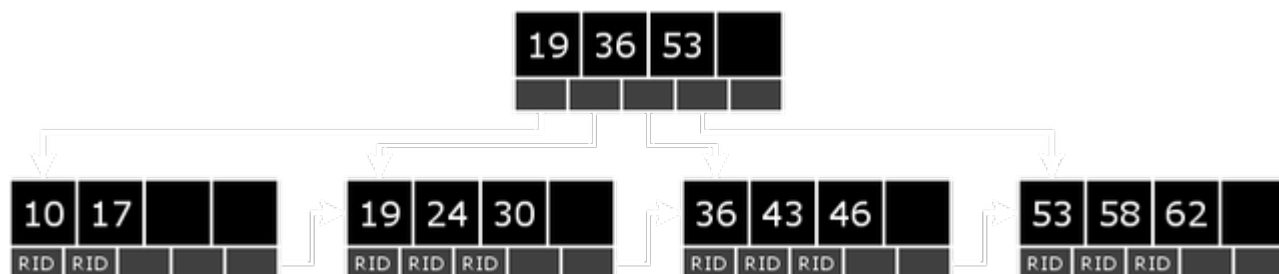
Select all Students, where this result is empty.

So in the end all students are selected where the course list has no courses with the exception of the taken courses.

Imagine there is a relation `PMoney(name:string, amount:integer)` and the SQL-Query `sql SELECT COUNT(*),COUNT(DISTINCT amount), MIN(amount), MAX(amount), SUM(amount), SUM(DISTINCT amount), AVG(amount), (VARIANCE(amount) + POWER(AVG(amount),2)) * COUNT(amount) AS 'E(X^2)' FROM PMoney` that results in `COUNT(*) | COUNT(DISTINCT amount) | MIN(amount) | MAX(amount) | SUM(amount) | SUM(DISTINCT amount) | AVG(amount) | E(X^2) | 8 | 5 | 0 | 6 | 15 | 14 | 2.5000 | 67` What is the result of the Query? `sql SELECT amount FROM PMoney ORDER BY PMoney.amount ASC` Enter the values in a comma separated list and use `NULL` for NULL values, e.g., `NULL,A,B,C` Don't use spaces between the elements, e.g, `A, B, C, NULL`

NULL,NULL,0,1,1,2,5,6

Consider the B+ tree below with the following properties: * Each node (except the root) contains $2 \leq k \leq 4$ keys. * Inner nodes: The keys in the subtree below the pointer `pi` are less than the key `ki`; the keys in the subtree below the pointer `pi+1` are greater or equal than the key `ki`. * Insert operations may only trigger node splits but not shifting of keys into bordering leaves. * When a node is split, the middle key is moved to the parent node. * Pointers `p1,...,p4` in leaves point to row IDs. Pointer `p5` in leaves points to the next leaf.  How does the B+ tree look like after insertion of the key 30?



Index structures support different kind of lookup operations. A point query looks up a single value. A range query looks up a range of ordered values. Which types of queries are supported by the following index structures: * Extensible hash table * B+ tree

- Extensible hash table → Point queries only
- B+ tree → Point and range queries

Consider the following two transactions: * T1: R(C); R(A); W(B); R(D); * T2: R(B); R(D); W(A); R(C); Which of the listed schedules is compatible with the 2PL protocol. * R1(C); R1(A); R2(B); R2(D); W1(B); W2(A); R1(D); R2(C); C1; C2 * R1(C); R1(A); R2(B); R2(D); W2(A); W1(B); R1(D); R2(C); C2; C1 * R2(B); R2(D); R1(C); W2(A); R1(A); R2(C); W1(B); R1(D); C1; C2 * R1(C); R1(A); W1(B); R1(D); C1; R2(B); R2(D); W2(A); R2(C); C2

R1(C); R1(A); W1(B); R1(D); C1; R2(B); R2(D); W2(A); R2(C); C2; *Serial schedule* \Rightarrow 2PL.

Consider a relation **R** holding information about the employees **e** in a large organization. Assume that the percentage of young employees (aged below 45) is 97% and the percentage of old employees (45 and above) is 3%. Let **L(e)** denote a single lock request, and **L(e)...** a series of identical lock requests of the lock type **L** on one or more employee tuples respectively. For each of the following database operations, select the most efficient sequence of locks to be acquired by the enclosing transaction T. You can assume that the transaction queries use an index on the employee's age to filter out old employees. `` *Choices: * X(R); * X(R); X(e)...; * S(R); X(e)...; * SIX(R); X(e)...; * IX(R); X(e)...; `` 1. Increase the salary of all young employees by 5%. 2. Compute the maximum old employee salary SY and reduce the salaries of the young employees when necessary, such that their salary does not exceed SY. 3. Increase the salary of all old employees by 5%. 4. Compute the maximum young employee salary SY and reduce the salaries of the old employees when necessary, such that their salary does not exceed SY.

1. Increase the salary of all young employees by 5%. $\rightarrow X(R);$
2. Compute the maximum old employee salary SY and reduce the salaries of the young employees when necessary, such that their salary does not exceed SY. $\rightarrow SIX(R); X(e)...$;
3. Increase the salary of all old employees by 5%. $\rightarrow IX(R); X(e)...$;
4. Compute the maximum young employee salary SY and reduce the salaries of the old employees when necessary, such that their salary does not exceed SY. $\rightarrow SIX(R); X(e)...$;

Consider the following three transactions: * **T1**: R(A); R(B); W(C); * **T2**: R(A); R(B); W(A); W(B); * **T3**: R(C); R(B); W(C); Which of the listed schedules is compatible with the 2PL protocol. 1. R2(A); R1(A); R2(B); W2(A); W2(B); R1(B); C2; W1(C); C1 2. R2(A); R1(A); R2(B); R1(B); R3(C); W1(C); R3(B); W2(A); C1; W2(B); C2; W3(C); C3 3. R1(A); R2(A); R2(B); W2(A); R1(B); W2(B); R3(C); C2; W1(C); C1; R3(B); W3(C); C3 4. R1(A); R2(A); R2(B); W2(A); R1(B); W2(B); W1(C); C1; R3(C); C2; R3(B); W3(C); C3

1. R1(A); R2(A); R2(B); W2(A); R1(B); W2(B); W1(C); C1; R3(C); C2; R3(B); W3(C); C3

Consider a DBMS buffer cache that can store four blocks. The buffer cache uses a LRU strategy which is implemented as a queue. In the beginning, the cache is empty. The blocks a, b, c, d, e, f and g are read in the following order: `a; c; e; g; e; a; f; d; e; b; f` In the table below, state the contents of the buffer cache after each block is read in the eviction order from right to left, i.e. the first entry to be evicted should be on the right. Example: adding e to abcd means that d will be evicted. Please note that your answers should not include upper case letters or spaces. You get 0.5 points for each correct answer. The contents of the buffer after reading the first three blocks are already given.

Accessed block	a	c	e	g	e	a	f	d	e	b	f
Buffer contents (queue)	a	ca	eca	geca	egca	aegc	faeg	dfae	edfa	bedf	fbcd

Quiz 2 - Cloud Computing & Distributed File Systems

Assume a cluster with 1000 machines. There are 300 machines of type X, 200 of type Y, and 500 of type Z. The probability of a machine to fail during the execution of a certain job is 0.0005 for X, 0.001 for Y and 0.00001 for Z. If a machine fails, the job fails. *Given that each machine runs a single job, compute the probability that the job will fail during execution. Give the number with 4 digit precision as answer and use '.' as fraction separator \rightarrow "0.1234". *

0.2989

$1 - ((1-p)^n * \dots)$

Cloud providers give availability guarantees in number of 9's. Calculate the maximum down time a system can have in one year for the given availabilities. Assume that a year consists of 365 days (no leap year, no leap second). Note the different units for the different availabilities. If the solution is a decimal number, please provide the number with 2 decimal points (e.g. 0.12, 1.58, etc.) * Availability guarantee of 90% * Availability guarantee of 99% * Availability guarantee of 99.9% * Availability guarantee of 99.99% * Availability guarantee of 99.999% * Availability guarantee of 99.9999%

- 876h
- 5256min
- 525.6min
- 3153.6sec
- 315.36sec
- 315.36sec

An user executes a look up on her EXT2 filesystem using a command on her Linux terminal and the following inodes and blocks are visited. What is that command? Suppose block 1 is the root directory. ![Alt text](https://i.ibb.co/gJHRqdC/image-9.png)

ls /aaa/eee

Consider a cluster that schedules jobs by using the **Shortest Task First Strategy**, which is implemented as a queue. In the table below, you can see the length and arrival of each job. Fill out the completion time for each job, as well as the order in which they are executed and calculate the average completion time. When calculating the average completion time, don't take into account the arrival time of the jobs, consider only the absolute completion time. Job|Length|Arrival ---|---|--- a|8|0 b|4|0 c|2|3 d|7|5 e|5|15 What is the order of execution and completion time for each job? What is the average completion time?

b 21, c 4, d 6, a 13, e 26

14

Consider a cluster that schedules jobs by using the **Round-Robin Strategy**, which is implemented as a queue. In the table below, you can see the length and arrival of each job. Assume a quantum of 30 ms. New incoming jobs will initially be placed at the head of the queue. Fill out the completion time for each job, as well as the order in which they are executed and calculate the average completion time. When calculating the average completion time, don't take into account the arrival time of the jobs, consider only the absolute completion time. Job|Length|Arrival ---|---|--- a|106|0 b|55|25 c|51|45 d|30|80 What is the order of execution? List the job for each quantum it runs, even if it does not run for the full 30ms. What is the average completion time?

a,b,c,d,a,b,c,a,a

183.25

Consider a scheduling problem with 3 jobs in a cloud environment. The cloud uses the dominant resource fair scheduling strategy to assign resources to jobs. Each job should be represented with at least one task. Job A's tasks each require 2 CPUs and 4 GB RAM. Job B's tasks each require 3 CPUs and 3 GB RAM. Job C's tasks each require 2 CPUs and 3 GB RAM. The cloud has a total of 18 CPUs and 24 GB RAM. *For each job, state if it is CPU or RAM dominant.* Assume a data center where this scheduling algorithm is dynamically applied. What is the maximum number of tasks that can be run for each job while still being fair and how big is the share of that job's dominant resource? The fraction should be provided as a number between 0 and 1 (e.g. 0.250 = 25%) with three decimal places. When all jobs have an equal share, the next resources get assigned in alphabetical order, i.e. A < B < C.

	Job A	RAM dominant	max tasks = 2	dominant resource fraction = 0.333
Job B		CPU dominant	max tasks = 2	dominant resource fraction = 0.333
Job C		RAM dominant	max tasks = 3	dominant resource fraction = 0.375

You want to determine the maximum size of you ext2 file system for a given block size. In your system, you have a block size of 1536 bytes (1.5KiB). Assume an ext2 file system as presented in the lecture with 12 direct pointers, 1 indirect pointer, 1 double-indirect pointer, and 1 triple-indirect pointer. Each pointer has a size of 32 bit. Provide your answer in gibibyte (10243) with 4 decimal places. How large can a single file be?

81.2115 GiB

Quiz 3 - MapReduce II & Key-Value Stores

Consider the two relations $R(r_1, r_2, r_3)$ and $S(s_1, s_2, s_3)$. The tuples of R have a size of 1100B, the tuples of S have a size of 128B. The cardinalities of R and S are: $|R| = 100'000$, $|S| = 100'000'000$. R and S are stored on 20 nodes in a cluster used for MapReduce. The data of both relations is distributed uniformly across all nodes. The number of mappers = number of reducers = number of nodes Mappers and reducers are co-located in the nodes Given is the following SQL query: ``sql SELECT * FROM R, S WHERE R.r1 = S.s1; `` Calculate how much data is transferred during a MapReduce broadcast join and a MapReduce partition join. You do not need to consider sending key-value pairs and simply assume that the record size accounts for this information. Partitioning is performed with a hash and modulo operation. Now assume there is filtering involved ($S.s_2 < X$) in the Map phase. The optimizer can chose between a partition-based and a broadcast join strategy. The amount of shipped data depends on the local predicate ($S.s_2 < X$). Compute the selectivity for which the broadcast strategy and the partition-based strategy transfer the same amount of data. Give the number with 4 decimal place precision. Remember: Selectivity is the fraction of tuples that match the filter predicate.

Cost of the *broadcast join*:

Send all data from R to all nodes. Each node has 5000 R tuples ($100'000 / 20$).

1100 bytes per record * (5000 tuple per node * 19 nodes to send the data to) * 20 nodes that send their data

==> 1100 byte * 5000 * 19 * 20 = **2.09 GB**

=====

Cost of partition join:

Partition R and S . Each node has 5'000'000 S tuples ($100'000'000 / 20$).

General form: record size * # nodes sending data * (# tuples per node / # nodes * (# nodes - 1))

Cost R: 1100 byte * 20 * (5000 / 20 * 19) = 0.1045 GB

Here, (5000 / 20 * 19) says that each node has to send one twentieth of their data to 19 other nodes. As we have 20 nodes, each of them does this, so 20 * (5000 / 20 * 19).

Cost S: 128 byte * 20 * (5'000'000 / 20 * 19) = 12.16 GB

==> total cost = cost R + cost S = 0.1045 GB + 12.16GB = **12.2645 GB**

=====

Selectivity:

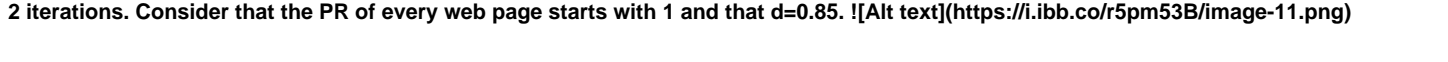
When we filter on S, the partition join sends less data. To find the selectivity that sets the cost of both joins about equal, we can put them into a short equation:

cost broadcast join = selectivity * cost S in partition join + cost R in partition join

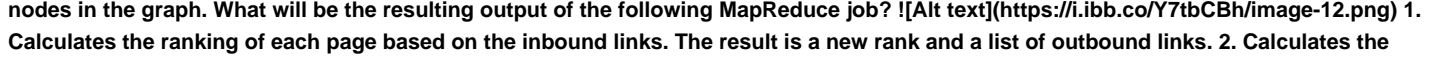
==> 2.09 GB = (x * 12.16GB) + 0.1045 GB

If we solve for x, we receive:

x = (2.09GB - 0.1045GB) / 12.16 GB = **0.1633**

In the graph below nodes are web pages and the directed edges are links to other pages. Calculate the Page Rank (PR) of web page C after 2 iterations. Consider that the PR of every web page starts with 1 and that d=0.85. 

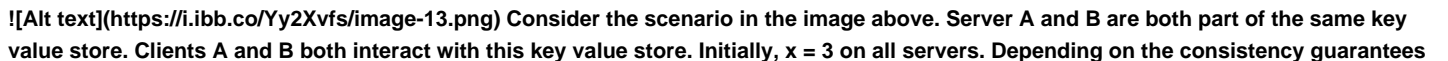
1.1818

You have a web page corpus at hand. It's constructed as a graph where each node has an ID, rank, and list of outbound links to other nodes in the graph. What will be the resulting output of the following MapReduce job?  1. Calculates the ranking of each page based on the inbound links. The result is a new rank and a list of outbound links. 2. Calculates the ranking of each page based on the outbound links. The result is a new rank and a list of outbound links. 3. Calculates the ranking of each page based on the inbound links. The result is a new rank of each page. 4. Calculates the ranking of each page based on the outbound links. The result is a new rank of each page.

1. Calculates the ranking of each page based on the inbound links. The result is a new rank and a list of outbound links.

According to the CAP Theorem, distributed systems can only guarantee two out of the following three properties: * Consistency - All nodes have the same view of the data at all times. * Availability - All requests sent to the system are answered. * Partition Tolerance - The system maintains its properties even in case of a network partition. This implies that we can build three different types of database systems CA, CP & AP. Based on this classification, which type of system would you aim for the following scenarios? 1. You are an online retailer which uses shopping carts to collect items that users want to buy. 2. You are a bank that wants to develop the software for its ATMs. 3. You are the owner of a social media application providing a social media feed.

1. AP
2. CP
3. AP

 Consider the scenario in the image above. Server A and B are both part of the same key value store. Clients A and B both interact with this key value store. Initially, x = 3 on all servers. Depending on the consistency guarantees given by the key value store, the read results of the clients as shown in the image can be different (shown as read x = ?). For the following consistency guarantees, select which value the x can be when reading. In some cases it might be possible that x can be multiple values. In that case explicitly select the option that states it can be "either ... or ...".

No guarantee: As the system provides no guarantees, it is possible that they read either 3 or 4, depending on whether the servers have synced their data.

Strong consistency: The system guarantees that there is only one valid version for a certain key at any given time. In this case, 4 will be the value that the clients read.

Monotonic reads: Monotonic reads only guarantee that if a process P has read a value for x, it will not read an older value in a new read. In this case, we cannot make any assumptions about the results of the reads that are performed for client B, so either 3 or 4 can be read on both clients. A reads his own writes.

Read your own writes: Client A is guaranteed to read 4, as it has just written it. Client B, on the other hand, has not written anything so it can receive either 3 or 4.

You have six nodes (in addition to the coordinator) and one transaction. How many messages are sent in total between the nodes and the coordinator to perform the **Two Phase Commit Protocol**? How many message round trips between a single node and the coordinator does the protocol require to complete in the success case?

24

2

Given is a distributed HashTable with an architecture similar to Chords: * Elements are mapped to a key within the range [0 ...1023]. * The nodes in the system are organized in a ring. * A node in the system has an id within the key-range. * Each node holds a fingertable with 8 entries. * An element is hosted by the first node whose id is equal or larger than the elements key. Given the following node setup: ![Alt text](https://i.ibb.co/H2TJR4Y/image-14.png) Node 60 receives a request for the element with the key 613. Provide *all* nodes that process the request.

60, 213, 358, 487, 591, 640

(242 is not included)

Quiz 4 - Stream Processing

Consider a stream processing engine that performs a window join on the tuple key. Assume a sliding window of length 10 seconds with a slide of 5 seconds. Time starts at 0, so the first window of length 10 ranges from 0 to 10, where 0 is included and 10 is not included [0, 10). You have two streams R and S with events in the form of (timestamp, key, payload). Stream R: { (1, a, 1), (3, a, 2), (7, b, 2), (8, c, 9), (9, a, 6), (12, b, 8), (13, c, 3), (16, b, 9) } Stream S: { (2, b, A), (4, c, D), (5, c, C), (7, a, K), (8, a, Z), (11, c, L), (13, b, N), (19, c, M) } You have received a watermark with the timestamp 22. Please specify how many tuples are in the output stream after joining up to the watermark.

Split into 3 windows.

Multiply for each window the count of elements per Stream

--> 21

Which of the following statements concerning Apache Storm, Flink and Spark are true? Select one or more: Apache Storm performs true streaming. All three engines support exactly-once guarantees. Apache Flink performs true streaming. Apache Spark performs true streaming. All three engines work on some form of execution graph

Apache Storm performs true streaming. Apache Flink performs true streaming. All three engines work on some form of execution graph

Consider a stream processing application using session windows to analyze some data. The session window has a defined gap of 5. Watermarks are generated by the processing engine based on a heuristic. Watermarks are used to trigger completed windows, i.e., watermarks are the only event type that produces output values. The stream is unordered and watermarks are used to indicate the maximum allowed lateness for events. The following stream consisting of events and watermarks is given as: (1, 8); (3, 2); (5, 4); w8; (14, 8); (12, 9); (18, 9); w16; w19; (24, 11); w24; (26, 17); w33 The events are composed of a timestamp t and a value v, so we have tuples of (t, v). A watermark with a timestamp i is given as wi. The events are processed in the order given above. State in the table below if a window is completed at watermark i. If so, calculate the sum of the values of the current window.

0

14

9

26

28

You are building a streaming application for the following use cases. Your goal is to have the best performance cost for each use-case. For each use case, decide which is the lowest processing guarantee that should be fulfilled for correct processing. Remember, the higher the guarantee, the higher the cost of processing. So the lowest guarantee is defined as the the guarantee with the lowest cost. For this exercise assume: Exactly-once > At-least-once > At-most-once. * You are a bank and all customer transactions are streamed from a central message queue through some normalization steps before they are persisted in the user's account-history log. * You are an online retailer that analyzes the content of their users' shopping carts to get an idea of which articles are frequently bought together. * You are an internet company that processes ad clicks and bills customer's based on the ad-click-event. Each click event is analyzed and the billing information is written to a key-value store with the event's id as the key.

Exactly-once At-most-once At-least-once

You want to build a stream processing engine and a thinking about how to efficiently store windows and which approach to use. To get a feeling for the storage cost, you consider a sliding window of length 1 hour with a 5 second slide. Your assumed stream has 1000 events per second. The options you are considering are naive tuple buffers and stream slicing. Using tuple buffers, you need one list of events per window and you assign each event to all the buffers that it belongs to. The slicing approach needs only one list of events per slice and you assign the event to the corresponding slice. An event is 120 bytes in size. If your application runs for 1 hour, how much storage do both

approaches require? For this calculation, you do not delete events. 1 MB = 10^6 bytes.

Slicing: $\#eventspersec * event\ size * time_in_sec = total_eventsslide_time\#eventspersec * event\ size = 432MB$

naive tuple buffer: $total_events * (time + window_size) / 2\#eventspersec * event\ size = 155736MB$

Select the true statement about mini-batches and true streaming. * Mini-batching has a higher latency and lower throughput than true streaming. * Mini-batching has a lower latency and higher throughput than true streaming. * Mini-batching has a higher latency and higher throughput than true streaming. * Mini-batching has a lower latency and lower throughput than true streaming.

Mini-batching has a higher latency and higher throughput than true streaming.

Select the true statement about time in stream processing. * Event time processing always uses a real-time timestamp. * Processing time is not a real-time timestamp. * Working with processing time can yield different results than working with event time. * Windows require processing time to be triggered. * Events always arrive ordered by event time.

Working with processing time can yield different results than working with event time.

Windows based on processing time can capture other events than ones based on event time, especially if some tuples arrive very late.

Quiz 5 - Graph Processing & ML Systems

You are performing a chain of matrix multiplications and want to know in which order to execute them for the best performance. For this calculation, you can assume the following number of FLOPS (floating-point operations) for a given matrix multiplication: $A_m * n \times B_n * p = 2 * n * m * p$ FLOPS Your matrix multiplications are given as: $A_{200*700} \times B_{700*300} \times V_{300*1}$ Please calculate the number of FLOPS for both orderings to determine which order is more efficient. $(A_{200*700} \times B_{700*300}) \times V_{300*1}$ FLOPS: $A_{200*700} \times (B_{700*300} \times V_{300*1})$ FLOPS:

do not forget *2 for the matrix multiplication (& +)

84120000

700000

For your ML system, you are thinking about how to represent matrices and are concerned with the storage cost. You are unsure whether to use a dense or sparse representation. The two options are a row-major dense matrix or a modified compressed sparse row matrix (MCSR). Your matrix cells store float values of 32 bit size. The sparse representation consists of an array of pointers (64 bit) to 2-array-data-structures for the index and value. The indices are stored as 32 bit integers and the values as 32 bit floats. Assume the data structure itself has no additional memory requirements, i.e. it is simply two arrays. Your matrix size is always 1200 x 900 (rows x columns). What is fraction of cells that need to be filled for the dense representation to have the same memory consumption as the sparse representation? Use four decimal point precision for the degree, e.g. 0.1234 (number between 0 and 1)

$matrix * 32bit = matrix.rows * pointer + (32+32)*x = \text{number of needed cells}$

$\text{number of cells} / \text{total number} = 0.4989$

You are designing a data parallel parameter server, and want to determine which update strategy to use based on the execution time. The workflow executes 3 batches of data on 3 workers. In the table below you are given the execution time of each batch per worker. For each update strategy calculate the total execution time for the update strategies listed below. Assume no time-gaps between batches. Batch 1 (s) Batch 2 (s) Batch 3 (s) Worker 1 20 35 28 Worker 2 27 32 30 Worker 3 25 38 33 Total execution time - BSP (s) Total execution time - ASP (s) Total execution time - Sync w/ Backup Workers - 1 backup worker (s)

Total execution time - BSP (s): 98

Total execution time - ASP (s): 96

Total execution time - Sync w/ Backup Workers - 1 backup worker (s): 90

Last steps cuts the slowest as well

Cypher: ```sql MATCH (subject:User {name:'Sarah'}) MATCH (subject)-[:WORKS_FOR]->(company:Company)-[:WORKS_FOR]-(person:User), (subject)-[:INTERESTED_IN]->(interest)-[:INTERESTED_IN]-(person:User) RETURN person.name AS name, count(interest) AS score ORDER BY score DESC ``` Given the following Cypher statement, select the correct corresponding SQL query. ```sql SELECT u1.name AS name, count(i1.topic_id) AS score FROM User u1 JOIN WorksFor w1 ON u1.id=w1.user_id JOIN WorksFor w2 ON w1.company_id=w2.company_id JOIN User u2 ON w2.user_id=u2.id JOIN InterestedIn i1 ON u1.id=i1.user_id JOIN InterestedIn i2 ON u2.id=i2.user_id WHERE u2.name = 'Sarah' AND u1.name != 'Sarah' GROUP BY u1.name ORDER BY score DESC; `` ``sql SELECT u1.name AS name, count(i1.topic_id) AS score FROM User u1 JOIN WorksFor w1 ON u1.id=w1.user_id JOIN WorksFor w2 ON w1.company_id=w2.company_id JOIN User u3 ON w2.user_id=u3.id JOIN InterestedIn i1 ON u1.id=i1.user_id WHERE i1.topic_id IN (`

```
SELECT i2.topic_id FROM User u2 JOIN InterestedIn i2 ON u2.id=i2.user_id WHERE u2.name = 'Sarah' ) AND u1.name = 'Sarah' AND u3.name != 'Sarah' GROUP BY u1.name ORDER BY score DESC; ```sql SELECT u1.name AS name, count(i1.topic_id) AS score FROM User u1 JOIN WorksFor w1 ON u1.id=w1.user_id JOIN WorksFor w2 ON w1.company_id=w2.company_id JOIN User u3 ON w2.user_id=u3.id JOIN InterestedIn i1 ON u1.id=i1.user_id WHERE i1.topic_id IN ( SELECT i2.topic_id FROM User u2 JOIN InterestedIn i2 ON u2.id=i2.user_id WHERE u2.name = 'Sarah' ) AND u3.name = 'Sarah' AND u1.name != 'Sarah' GROUP BY u1.name ORDER BY score DESC; ```
```

First one does not match interests to user Other would return Sarah

```
SELECT u1.name AS name, count(i1.topic_id) AS score
FROM User u1 JOIN WorksFor w1 ON u1.id=w1.user_id
      JOIN WorksFor w2 ON w1.company_id=w2.company_id
      JOIN User u3 ON w2.user_id=u3.id
      JOIN InterestedIn i1 ON u1.id=i1.user_id
WHERE i1.topic_id IN (
      SELECT i2.topic_id
      FROM User u2 JOIN InterestedIn i2 ON u2.id=i2.user_id
      WHERE u2.name = 'Sarah'
) AND u3.name = 'Sarah' AND u1.name != 'Sarah'
GROUP BY u1.name
ORDER BY score DESC;
```

10 Stammdatenmanagement

Was versteht man unter Informationsqualität ?

Informationsqualität ist das Maß der Übereinstimmung von Daten und den realen Objekten, die diese repräsentieren.

Was sind Stammdaten ?

Stammdaten repräsentieren reale Geschäftsobjekte mit dem Ziel, die Prozesse mit den benötigten Daten/Informationen zu versorgen. Sie bilden die fundamentale Grundlage einer Organisation. Sie werden normalerweise selten geändert, sind eindeutig Nummeriert und haben ein kontinuierlichen Datenvolumen.

Welche grundlegenden Aufgaben/Tätigkeiten gibt es bei der Datenhaltung ?

- Erfassen
- Verarbeiten
- Pflege
- Abfragen (Auffinden)

Die Operation Auffinden bestimmt häufig über das Antwortzeitverhalten des gesamten Informationssystems.

Was ist Stammdatenmanagement und wieso ist es wichtig ?

Ist das Management zur Sicherstellung der Qualität der Stammdaten. Verfolgt den Zweck die Eignung der Stammdaten für alle weiteren Prozesse im Unternehmen zu Gewährleisten. Aufgaben sind die Datenintegration, Datenqualität und Datenkonsolidierung.

Stammdatenmanagement ist kein reines IT-Problem. Viel mehr zieht es sich durch sämtliche Geschäftsbereiche und ist für den gesamten Geschäftserfolg von Relevanz.

Welche Potentiale und Risiken gibt es bei Stammdaten ?

Gute Stammdaten

- optimale Versorgung aller Beteiligten
- Prozessautomatisierung
- Prozesssteuerung

Schlechte Stammdaten

- Fehler in Prozessen
- Fehler auf Belegen
- Fehlerhafte Auswertungen

Welche Herausforderungen gibt es im Stammdatenmanagement ?

- Unterschiedliche Terminologien verstehen

- Qualitätsanforderungen erfüllen (z.B. Konsistenz)
- Veraltete Architekturen der Anwendungssysteme
- Redundante Datenhaltung
- organisatorische Bedeutung
- Definition und Kontrolle der Stammdatenmanagementprozesse

Welche Kriterien/Einflussfaktoren wirken sich auf die Stammdatenqualität (nach `Apel`) aus ?

- Korrektheit und Fehlerfreiheit
- Konsistenz
- Aktualität und Gültigkeit
- Vollständigkeit
- Redundanzfreiheit
- Einheitlichkeit und Eindeutigkeit
- Prozesse des SDM
- Verwendete Datenquellen

Welche Aufgabengebiete kennt das Qualitätsmanagement für Stammdaten ?

- Datenanalyse: Identifikation von Problemen, Graphische Analysen, Plausibilitätslisten, Statistische Auswertungen
- Datenanreicherung: Externe Referenzdaten, Mehrsprachigkeit, Maßeinheiten
- Datenbereinigung: Doublettenerkennung, Rechtschreibkorrektur, Plausibilitätsprüfung

Was ist MDM ?

Master Data Management. Ist ein Synonym für Stammdatenmanagement.

Wie könnte ein Beispiel für Stammdatenmanagement in einem KMU aussehen ?

Schlechtes altes ERP System und keinerlei Wissen zu Stammdatenmanagement haben zu Redundanzen, Unvollständigen und nicht Verständlichen Stammdaten geführt.

Einführung eines Stammdatenmanagers und eines Beauftragten pro Abteilung. Schulung, Beratung und Sensibilisierung.

Stück für Stück einen Stammdatenmanagement Prozess entwickeln und am Ende tolle Stammdaten haben.