CMI/Jan-Apr-2020/End-Semester Exam Aug 2020

Chennai Mathematical Institute

Big Data with Hadoop Deadline: Section I: 1 PM Section II: 5 PM.
Max Marks: 35 (for Section I) + 20 (for Section II).

Roll No.: _____ Date: Aug 2020

Name: _____

Instructions

- This is an open book test.
- Calculators are allowed.
- Please do not mail your answers to your instructor. Upload your answers (as a single PDF file for each Section) to Moodle. Name your pdf as <rollno>-<fname>-sec1.pdf and <rollno>-<fname>-sec2.pdf. If the upload fails, mail your work to venkateshv@iiitd.ac.in with subject line "BDH Exam".
- You have 4 hours to complete Section 1. You have an additional 4 hours to complete Section 2.
- No negative marks for any section.
- You may use internet resources. Do not discuss with friends. Do not engage in any kind of plagiarism.

Section 1: Questions 1 - 5 carry 7 mark each.

Question 1. Draw a sequence diagram to explain the process of updating free space bitmap. In your own words, explain the sequence diagram.

Solution A sequence diagram depicts the interaction between instances of objects in a system. Here, we consider the interactions between the user and the disk through the file system. Finding suitable abstractions helps us manage the complexity of interactions. There are many ways to represent the interactions. Here is one way where we use four instances as suitable abstractions for drawing.

When the user (an instance of whom is represented as *:User*) writes a block, the operating system (represented as *:OS* through the file system component) needs a free block address to place the block on the disk (represented as *:Disk*). Let us say, this component is called the Free Space Manager referred to as *FreeSpaceMgr* in Figure 1. The Free Space Manager need to access the bitmap from the disk, update it to reflect that this block is now updated with the to-be written block. Once it has the address, it may write to the new block. This is just one part of the sequence diagram. Similarly, deletion of a block can be represented in a sequence diagram.

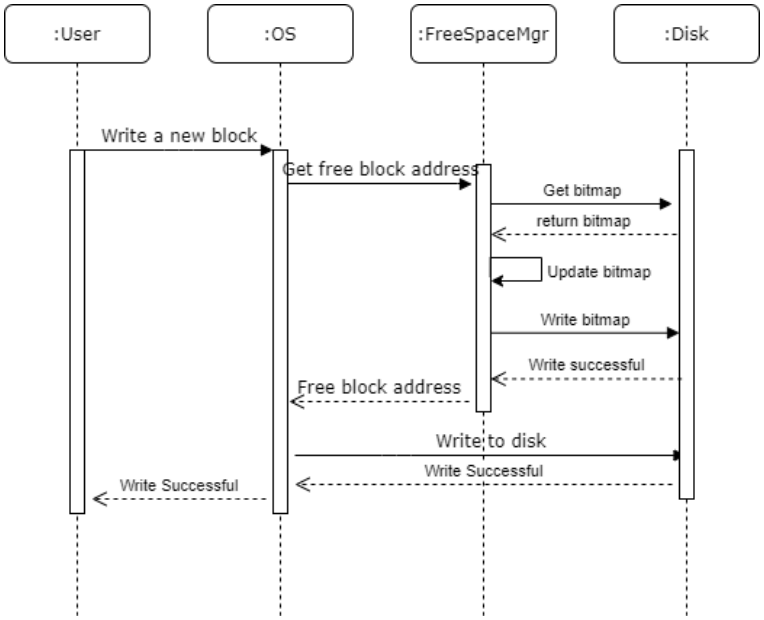


FIGURE 1. Sequence Diagram

Question 2. A distributed computing environment uses global vector time stamps as discussed in the class. You are provided with the following facts:

(1) $(1,1,1,1) \rightarrow (2,1,1,1)$ is a happens-before relation.
(2) Exactly two events occurred in each process.
(3) Between the two events of every process, at least one event occurred in another process.
(4) If the last digit of your roll number is r, the first event occurred in the process p_i where $i = (r\%4) + 1$.

Draw the hasse diagram and list the global vector time stamps. Note that there may be more than one hasse diagram agreeing to the above facts. You may draw any one.

Solution

The dimensions of the happens-before vector gives away the number of processes. With four processes, Figure 2 gives one possible interaction between our distributed processes. Note that we follow a global vector time stamps here. Depending on your roll number, instead of p_1, the first event will occur somewhere else. Also, note that the $(1,1,1,1) \rightarrow (2,1,1,1)$ is a happens-before relation in the diagram. The second event happens first on p_1.

In our construction, we have a total order. Therefore, the hasse diagram is just a straight line as dictated by the timestamps.

Question 3. Given any vector $z = (z_1, z_2, ...z_K) \in \mathbb{R}^K$, its distinctly valued components (z_i) may not add up to 1. There are many ways to transform z into another vector such that this property is achieved. Here are two ways:

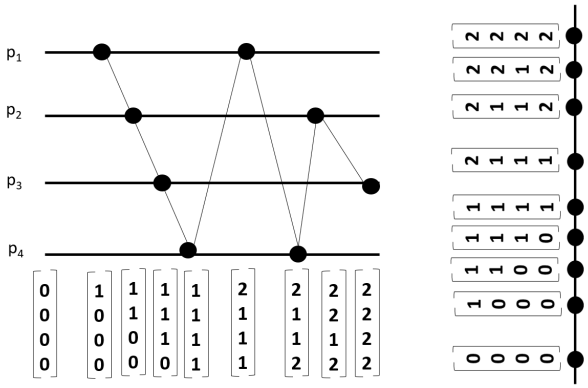


FIGURE 2. Global Vector Time Stamped on Processes and the Corresponding Total Order of Events

(1) Use $argmax(z)$ where the component with maximum value gets 1 and all others get 0. For example if $z = (1, 2)$, our new representation $argmax(z) = (0, 1)$.
(2) We can use softmax function, $\sigma(z) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$.

Assuming we have a large number of such fixed length vectors, provide map-reduce design(s) to calculate both the argmax and softmax values. The input vectors are presented as a HDFS file with each line containing a vector. You are required to get the output stored as two separate files, one for argmax vectors and the other for softmax vectors. You need not code. Draw and explain the map-reduce design.

Solution In the simplest design, a mapper is sufficient to convert a vector to its argmax and softmax representation. The same mapper can write two files as output. This is because we don't need more than one line from input to arrive at the output. Figure 3 shows the idea pictorially. A "Filter" is not a suitable term here as we don't filter any row. We just process it.

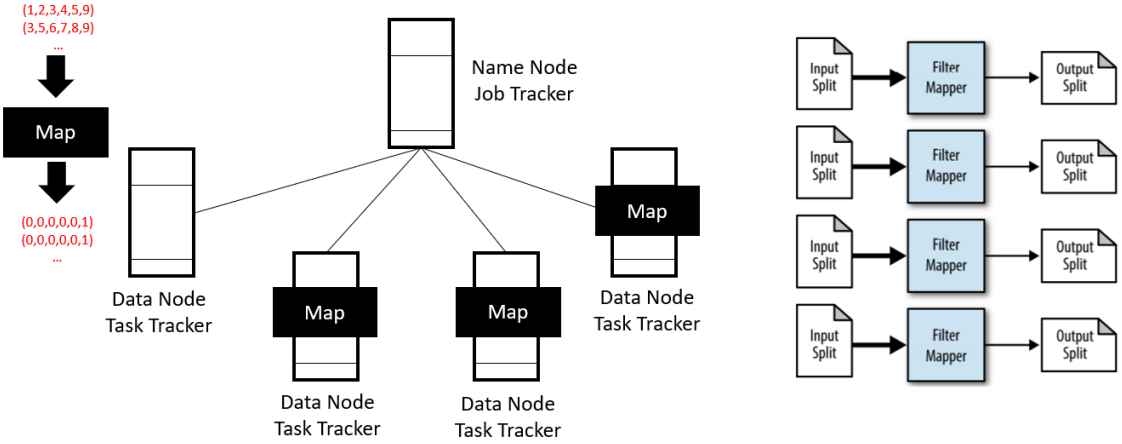


FIGURE 3. A Simple Map Reduce Pattern

Question 4. You are hired as a DB administrator for Chunnalal Travel Agency (CTA). CTA is building a ticket reservation web service. CTA faces the following challenges and looks up to you for suggestions.

(1) CTA recently recruited staff skilled in key-value datastores. Identify two scenarios where a key-value datastore can by used by CTA.
(2) If every ticket ever booked through the service is logged for future reference, which type of datastore will you use and why?

Solution Key-Value stores have a reputation of being very fast for reads and writes. Another advantage is that the storage overhead is minimal. There are more advantages. Yet, these two lead to the following use-cases:

(1) Real-time processing: CTA can introduce a real-time "number of users logged in" ticker to show that their site is popular.
(2) Caching to enable other web services: As we saw that redis is used in the load balancing use case, CTA could store information such as "trains running discounted fare" for search related to reservation.

Several kinds of log formats, the volume of logs and the write-once nature of logs make NoSQL datastores perfect candidates for their storage. We discussed, key-value, graphdb, document db and columnar db in class. If logs can be structured such as say, *ID, timestamp, log string, Additional Data*, then we could consider a columnar db. As additional data grows or becomes more structured, columnar db gives us an advantage of querying them effectively. We also saw JSON based semi-structured datastores that could also serve the same purpose. Here, it is important to understand if we prefer consistency, partition-tolerance or availability over one other. We can achieve two of the three. Since logs are not sensitive for consistency, a datastore that prioritize for AP such as Cassandra or CouchDB would be more appropriate.

Question 5. CMI wants to build a web service which provides a knowledge repository such as Wikipedia for explaining the concepts of algebraic geometry. You may select any small part of this system (for example, adding a concept, accepting comments, etc) to illustrate how to design a web service. Design a REST based web service for this purpose and through that scenario, explain the role of non-idempotent REST methods.

Solution Designing a REST service involves four key steps which we detail here. This solution hint considers "Adding a concept" scenario. It can be extended for any such scenarios.

(1) Identify the object model: concept
(2) Create Model URIs: /concept/{ID}
(3) Determine Representations: An XML representation is shown below.
(4) Assign HTTP Methods: POST is a non-idempotent method. A post call will change the state of the system. We use it here to create a new post as in "HTTP POST /concept". This takes the representation XML of concept as data payload. Server creates the concept on receiving a post request. It could create an entry in its database and return a "HTTP 200 OK" response if everything is successful.

A sample representation of a concept:

```
<concept id=1>
    <name>Algebraic Variety</name>
    <description>
    An algebraic variety is defined as the set of
    solutions of a system of polynomial equations over the real
    or complex numbers.
    </description>
</concept>
```

Section 2: Questions 6 - 9 carry 5 mark each.

Question 6. Some students at CMI wanted to publish the courses offered in a particular semester, its pre-requisite courses, the instructor, teaching assistant and such details as a neo4j graph. Can you setup a part of this graph using neo4j sandbox/desktop? Provide the neo4j graph, the neo4j commands (and a sample of data if any) that would create such a graph. Your graph must have at least 7 nodes and 12 nodes as the maximum.

Solution The graph shown in Figure 4 can be generated using the following commands:

```
//Lets clean up everything.
match (n) detach delete n;

//Setup the graph. A semicolon helps you to run multiple queries together. If
     this does not work, enable the settings appropriately.

//Creating pre-requisites
create (c1:Course{name:'DMML'})-[:IS_A_PREREQUISITE_OF]->(c2:Course{name:'AML
    '});
match(c2:Course{name:'AML'}) create (c1:Course{name:'NLA'})-[:
    IS_A_PREREQUISITE_OF]->(c2);
match(c2:Course{name:'NLA'}) create (c1:Course{name:'LAA'})-[:
    IS_A_PREREQUISITE_OF]->(c2);

//Creating teaches
match (c2:Course{name:'DMML'}) match (c1:Course{name:'AML'}) create (a:
    Instructor{name:'Madhavan'})-[:TEACHES]->(c2) create (a)-[:TEACHES]->(c1)
    ;
match (c2:Course{name:'NLA'}) match (c1:Course{name:'LAA'}) create (a:
    Instructor{name:'Kavita'})-[:TEACHES]->(c2) create (a)-[:TEACHES]->(c1);

//Creating offered-in
```

```
match (c2:Course{name:'NLA'}) match (c1:Course{name:'LAA'}) create (c2)-[:
    IS_OFFERED_IN]->(a:Semester{name:'Sem2'}) create (c1)-[:IS_OFFERED_IN]->(
    b:Semester{name:'Sem1'});
match (c2:Course{name:'DMML'}) match (c1:Course{name:'AML'}) match (e:
    Semester{name:'Sem2'}) create (c2)-[:IS_OFFERED_IN]->(e) create (c1)-[:
    IS_OFFERED_IN]->(b:Semester{name:'Sem3'});

//Creating is_ta_for
match (c2:Course{name:'NLA'}) match (c1:Course{name:'LAA'}) create (a:
    Assistant{name:'Abhishek'})-[:IS_TA_FOR]->(c1) create (b:Assistant{name:'
    Anubhab'})-[:IS_TA_FOR]->(c2);
match (c2:Course{name:'DMML'}) match (c1:Course{name:'AML'}) create (a:
    Assistant{name:'Debjit'})-[:IS_TA_FOR]->(c2) create (a)-[:IS_TA_FOR]->(c1
    );

//See the graph
match (n) return n;
```

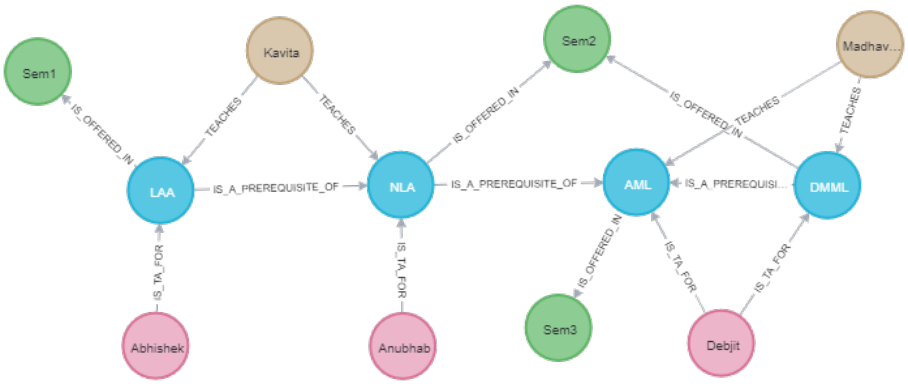


FIGURE 4. Neo4j Graph

Question 7. Design an Apache Storm topology for softmax and argmax computation as explained in Question 3. For this purpose, draw the topology diagram and explain the streams, spouts and bolts. You do not need to write any code.

Solution Typically storm topologies are used to setup realtime computation. Here, let us assume that the conversion is required to support some other realtime application. Another possible usecase is that the HDFS stores are different.

In a storm topology, Stream is an unbounded sequence of tuples. Spouts create the streams. Spouts connect to the source such as web servers or HDFS (as in this case) to pull data. Here, we have only one kind of intput. But, it requires two different kinds of processing. Hence, we create two bolts, one each to process argmax and sosftmax respectively. Technically, it is possible to have the same bolt do both. The separation helps us in future-proofing our design, just in case, we extend these ideas into many more complex

computations. Bolts process the streams and emit new streams. Since we need all records from spout to go to both the bolts, we use "All" grouping.

Spouts and Bolts run in parallel. Storm guarantees no-data-loss even if some nodes go down while processing. Moreover, it allows us to implement flow-control. With flow-control, we can decide the maximum data that can be queued up as tuples on stream for a bolt. Figure 5 shows two topologies, one with a single HDFS Bolt and all-grouping, and another topology with multiple HDFSBolts with shuffle-grouping.

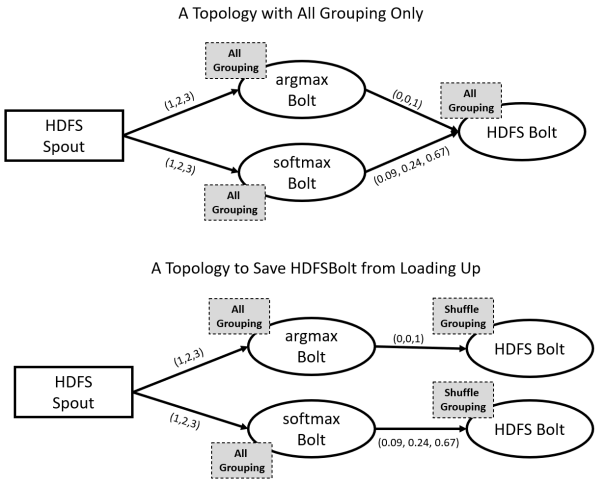


FIGURE 5. Two topologies with different grouping strategies.

Question 8. GRE and GMAT exams are conducted in computer-based testing formats. Students visit individual testing centers in their local town to take the exam. As and when the student completes an exam, each system in this testing center submits the test data to either of the two web services namely the GRE and GMAT web services. These web services not only capture the test set details, the student solutions, but also capture the other data such as time spent by each student on each question, the number of times a particular question was revisited and the order of answering the questions. We wish to save these data in two different HDFS stores, a) one for storing the results, b) another for rest of the data for research on how to improve the tests.

(1) Draw and explain a multi-agent data flow model with flume for the above-mentioned scenario. (3 marks)
(2) Within any part of the given scenario, identify a flume event and explain its contents with examples. (2 marks)

Solution Given that there are two HDFS stores to save the results to, we configure two sinks. Similarly we have two sources subscribed to two different data elements. So, a simple multi-agent flume data flow model will look like Figure 2;

The role of flume source is to receive events and store them in one or more channels. Events stay in channels until flume knows that the event is moved by the sink successfully.

Sink removes the event from channel after its job is over. So, the channel capacity plays a major role in the reliability of the system.

Figure 6 shows a flume multi-agent data flow model.

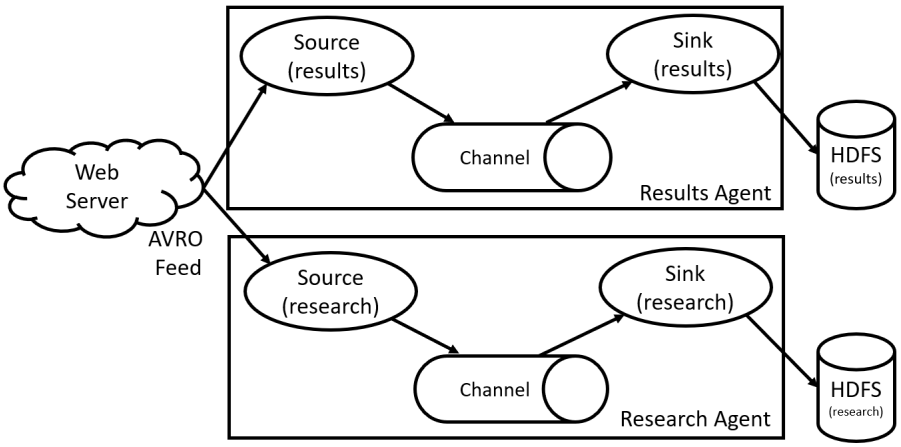


FIGURE 6. Flume Data Flow Model

A flume event is the unit of data flow containing a payload and optional attributes. We could have the test results published as a Flume Event as follows: TestTakerID, Score. As a concrete example, data records may flow as follows: {(VEN1000202, 98), (MMD000203, 87), ... }. Also, assume some attributes are set in the input to distinguish between results and the rest. Say, an attribute "type" has a value "TestScore", then our results agent is interested in it. This can be represented in a JSON format for which an AVRO schema can be created as shown below.

```
{
    "type": "string",
    "TestTakerID": "string",
    "score": "int"
}
```

Once the schema is available, this data can be serialized in binary format and used by flume. Both the data producer and the flume (consumer) must have the same AVRO schema to work with the data feed.

Question 9. CMI was excited to go digital by streaming its annual day festival events. Specifically, it published the results of all the events in real-time. Three news TV channels decided to subscribe to these events and show them as running news repeatedly throughout the day. Draw a kafka-based publish subscribe architecture to implement this use-case. Take any example within the scope of the given use-case to explain *topics* and *partitions*.

Solution Some key facts to know about kafka[1] are:

[1]See https://kafka.apache.org/intro for more information.

(1) If a producer publishes two events with same topic and to the same partition, Kafka ensures that any consumer reading these events will see the events in the same order.
(2) Events are organized and stored as topics.
(3) Every new event published is appended to one of the topic's partitions.
(4) If the publisher does not specify the partition ID, the message key is used to decide the partition.

Let us assume that CMI festival publishes results of on-going chess and coding competitions. We have one producer publishing events to two topics namely chess and coding competition. There are three consumers, the three TV channels. Lets call them SunTV, StarTV and MoonTV. To improve performance, you may increase the number of Kafka brokers (servers that store the partitions are called brokers) as and when necessary. So, the architecture looks as shown in Figure 7.

All CMI writes are organized and stored into the two topics. Each topic is configured to have p partitions. In this example, CMI writes pairings and results data for chess competition with these keys. So, with a configuration of two partitions for the Chess topic, kafka assigns each key to a partitions. Since the partitions may reside in different servers (brokers), this allows for parallelism. Increasing the partitions improves throughput because the consumers can access them in parallel.

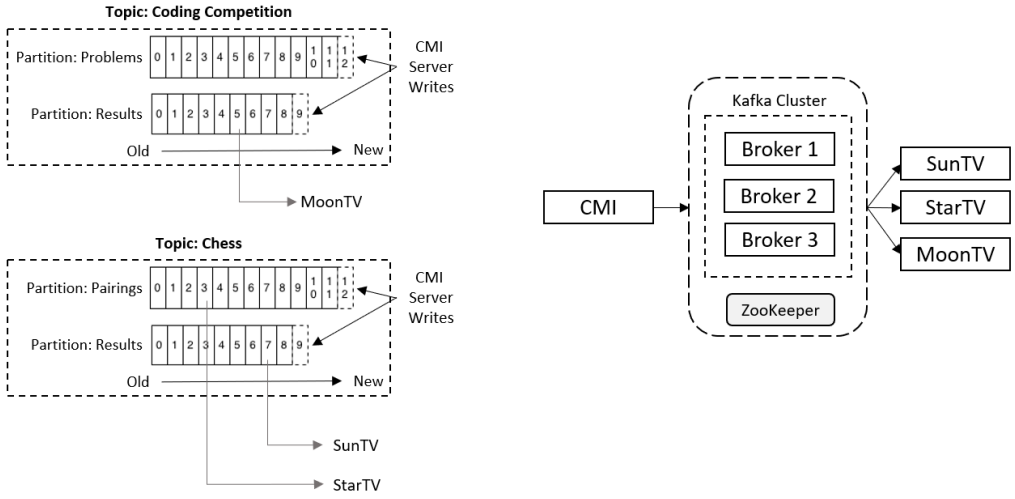


FIGURE 7. Kafka Architecture for CMI Events

CMI/2021/Final-Semester Exam July 2021

Chennai Mathematical Institute

Distributed Computing and Big Data

DURATION: $2\frac{1}{2}$ HOURS + 30 MINS FOR UPLOADING. MAX MARKS: 35.

ROLL NO.: _____ DATE: 07/07/2020

NAME: _____

Instructions

- Submit a single pdf file carrying your answers on moodle under "Final" assignment. For any reason, if you cannot upload to moodle, email your work to vvtesh.cmi@gmail.com.
- This is an individual assessment. Do not discuss with anyone.
- Please stop writing after $2\frac{1}{2}$ hours.
- Please remember to mention your name and roll number in your answer sheet.
- Late submissions will attract penalty.

Section 1: All questions carry 5 marks each.

Consider the following version of the problem of muddy children puzzle.

Suppose that there are a total of N children, where M ∈ {1, ... , N} of them has mud on their forehead. Each child can observe whether another child (but not himself or herself) has mud on their forehead. The muddy children protocol goes in rounds. Before the first round, the father declares that there is at least a muddy child (i.e. with mud on their forehead). In each round, he asks the children whether they know if they are muddy, to which the children can answer yes or no. The children are perceptive, intelligent, and truthful.

Question 1. Answer the following:

(1) If $N = 4$ and $M = 2$, enumerate what each child would have mentioned in each round till all the responses converge. For each round, clearly explain the reasoning behind every child's answer. (2.5 Marks)

(2) If $N = 4$, $M > 0$, and M is even, enumerate what each child would have mentioned in each round till all the responses converge. For each round, clearly explain the reasoning behind every child's answer. (2.5 Marks)

Section 2: All questions carry 5 marks each.

The Ministry of Health and Family Welfare (MoHFW) needs to maintain the COVID vaccination status. Therefore, it requires the hospitals and other camps where vaccines are administered, to collect and upload the data related to the person who is getting vaccinated.

Question 2. Design a RESTful web service for maintaining COVID vaccination records for the Ministry of Health. You may scope your answer to two resources. Your answer must cover at least one idempotent method assignment and one non-idempotent method assignment.

Question 3. Explain a strategy for hospitals to securely transfer the vaccination data through web services using a sequence diagram.

Question 4. Developers nowadays commonly make use of Object-Relation Mapping (ORM) frameworks to provide a conceptual abstraction between objects in Object-Oriented Languages and data records in the underlying database. Provide a partial class-diagram with at least two classes for the above mentioned MoHFW scenario. Explain how ORM can be used in this scenario.

Section 3: All questions carry 5 marks each.

Question 5. An N x N adjacency matrix describes if a person is a friend of another person. You may visualize this as a directed graph where nodes represent users and edges represent friendship. Note that X is a friend of Y does not mandate that Y needs to be a friend of X. We want to associate an influencer score for each user. The score is any integer directly proportional to the number of users that they are connected to, either directly or transitively. Assuming that the matrix is very large, explain a *map-reduce design pattern* to accomplish this task.

Question 6. To implement the vector clocks, we need to fix the number of bits for each integer element of the vector. If the bit count is too large, we need more space to store them and if the count is too small, we may end up in a overflow situation. For example, a two bit clock implementation can only describe three events (1, 2 and 3. Recall that 0 is needed as a special case) in a process. Assuming that each process in our distributed systems may have several billions (or even more) events, how can we decide the optimal size for the elements of a vector clock implementation?

Question 7. Provide distributed algorithms for the following:

(1) To elect two processes as coordinators from a group of n independent processes. (2 marks)

(2) To elect m processes as coordinators from a group of n independent processes where $m < n$. (3 marks)

CMI/2022/Final-Semester Exam May 2022

Chennai Mathematical Institute

Distributed Computing and Big Data

DURATION: 2 HOURS MAX MARKS: 35.

ROLL NO.: _____ DATE: 12/05/2022

NAME: _____

Instructions

- Please remember to mention your name and roll number in your answer sheet.
- You are allowed to bring one handwritten A4 sheet with notes. You will not be allowed to borrow or lend notes from or to other students.
- No electronic devices (calculators, laptops, etc) are allowed in the exam hall.
- No negative marks.

Section 1: All questions carry one mark each.

Question 1. Which of the following depicts the correct order by the factor of increasing size? Choose the best answer.

(1) petabytes, exabytes, zettabytes
(2) exabytes, petabytes, zettabytes
(3) exabytes, zettabytes, petabytes
(4) terabytes, exabytes, petabytes

Question 2. To store an object in Amazon S3, you create a bucket in one of the AWS regions and then upload the object to it.

Question 3. Which CAP category does Apache Cassandra belong to? AP

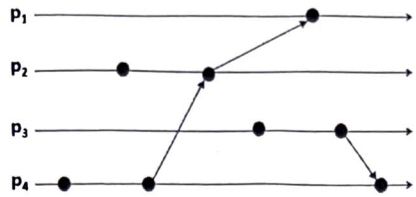
Question 4. Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications. True/False?

Question 5. Azure regions are physically separate locations within each Azure availability zones that are tolerant to local failures. True/False?

Section 2: All questions carry three marks each.

Question 6. If only 80% computation can be executed in parallel, and if we have 8 processors, what is the best speed up achievable as per Amdahl's law?

Question 7. Annotate the following space-time diagram with vector time.



Question 8. CMI was excited to go digital by streaming its lectures on its own platform. Three other universities decided to subscribe to these lectures. Draw a kafka-based publish subscribe architecture to implement this use-case. Take any example within the scope of the given use-case to explain *topics* and *partitions*. You are not expected to write any code.

Question 9. What will be the output of the following pig script if the input contains a single line of $(1,3,5)(2,4,6)$?

```
A = LOAD 'data' AS (
 t1:tuple(t1a:int, t1b:int,t1c:int),
 t2:tuple(t2a:int,t2b:int,t2c:int)
 );
X = FOREACH A GENERATE t1.t1a,t2.$1;
DUMP X;
```

Question 10. Assume a disk size of 4 Terabyte with block size of 4 KB. How much space will you need to store the free space bitmap?

Section 3: All questions carry 5 marks each.

Question 11. Thomas Cook is a travel management consultancy. In the post-covid era, they want to automate their travel booking systems. Design a RESTful web service for the travel booking scenario. You may scope your answer to two resources. Your answer must cover at least one idempotent method assignment and one non-idempotent method assignment.

Question 12. Describe a map-reduce design pattern to join two tables from an RDBMS.

Question 13. Provide distributed algorithms for the following:

(1) To elect three processes as coordinators from a group of n independent processes. (2 marks)

(2) To elect m processes as coordinators from a group of n independent processes where $m < n$. (3 marks)

CMI/22/4/2023/Final Exam Apr 2023

Chennai Mathematical Institute

Distributed Computing and Big Data

DURATION: 3 HOURS MAX MARKS: 35.

Instructions

- Please remember to mention your name and roll number in your answer sheet.
- This is an individual task. Do not discuss with anyone.
- This is a closed book exam. You are not allowed to carry books or cheatsheets.
- No electronic devices (calculators, laptops, etc) are allowed in the exam hall. Wherever heavy calculation is involved, you need not evaluate it to the final number unless it is explicitly asked for. For example, it is acceptable to leave the answer as $\frac{1}{1+\frac{5}{32}}$. You need not evaluate it to 0.865.
- First section has negative marks. No negative marks for the rest of the sections.

Section 1: All questions carry one mark each. -0.5 for wrong answers. Answer in True/False.

Question 1. The name notwithstanding, there are most definitely servers in serverless computing. 'Serverless' describes the developer's experience with those servers—they are are invisible to the developer, who doesn't see them, manage them, or interact with them in any way. True

Question 2. Poorly maintained data lakes are often called Data Swamps. True

Question 3. Apache Kafka is an open-source distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications. True

Question 4. For the services that do not expose metrics, we can use the ladder based scaling strategy. Scaling ladders can be defined per million concurrent users on the platform (1M, 2M ..). This works well for predictable workloads. True

Question 5. A Content Delivery Network (CDN) is a distributed network of servers that are geographically distributed across the globe. True

Question 6. Load balancer is a device or software that distributes incoming network traffic across multiple servers. True

Question 7. Pods are the smallest deployable units of computing that you can create and manage in Kubernetes. True

Question 8. Neo4j is not ACID compliant. False

Question 9. Redis is a key-value store and MongoDB is a document Store. True

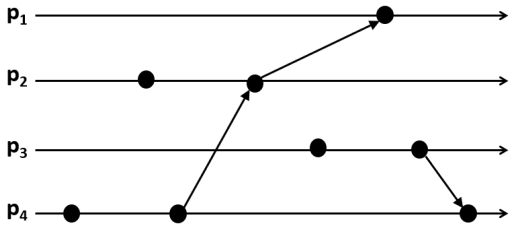
Question 10. On the CAP triangle, MongoDB falls on the AP side and Cassandra falls on the CP side. False

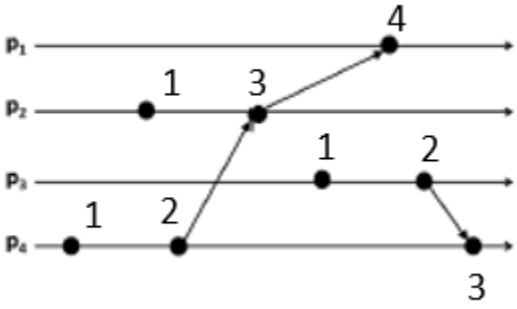
Question 11. General-purpose computing on graphics processing units is the use of a GPU, which typically handles computation only for computer graphics, to perform computation in applications traditionally handled by the CPU. True

Section 2: All questions carry two marks each.

Question 12. If only 10% computation can be executed in parallel, and if we have 10 processors, what is the best speed up achievable as per Amdahl's law?
R = 90%, P = 10. Best speedup = $\frac{1}{\frac{90}{100}+\frac{1-90/100}{10}} = \frac{100}{91} = 1.0989$

Question 13. Annotate the following space-time diagram with scalar time.





Question 14. For the same space-time diagram as in the previous question, annotate p_2 events with matrix time.

$$p_2^1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad p_2^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

Question 15. What will be the output of the following pig script if the input file, file1, contains a single line "1,V Rao,40,Chennai"?

```
A = LOAD 'file1' USING PigStorage(',')
    AS (id:int, name:chararray, age:int,  city:chararray);
B = FOREACH A GENERATE TOKENIZE(name);
DUMP B;
```

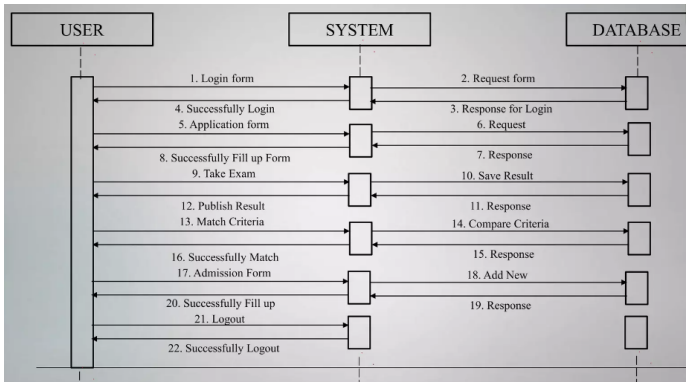
Solution: ((V),(Rao))

Question 16. What will be the output of the following pig script if the input file, file2, contains numbers 1 to 10, each in one line (i.e., 1 in first line, 2 in second line, and so on)?

```
A = Load 'file2' using PigStorage(',') as (num:int);
B = Foreach A generate 1 as gid, num;
C = Group B by gid;
D = Foreach C generate SUM(B.gid);
Dump D;
```

Solution: (10)

Question 17. Draw a sequence diagram to capture the CMI's admission process. Should have at least two objects, swim lanes, and life lines. A sample solution is here.



Question 18. Ramesh bought a hard disk with rotational delay of 3ms. With what RPM does the disk spin? If it had 20 sectors per track, what is its read time?

Solution: (10,000 RPM. Read time $= \frac{6}{20} = 0.3$ms)

Question 19. How many nodes are created when the following statement is executed by Neo4j?

```
create (p:Person {name:'Venkatesh'})-[:Teaches]->(c:Course {name:'BigData'})
```

Solution: 2 nodes

Section 3: All questions carry 4 marks each.

Question 20. The Chennai Public School wants to automate its system for grading students. Specifically, this system will allow creation, modification and deletion of exam marks and student grades from class V to class X. Design a RESTful web service for this scenario. You may scope your answer to three identified resources. Your answer must cover at least one idempotent method assignment and one non-idempotent method assignment.

Question 21. From a very large text file, we need to find the least five frequent words that contain only alphabets (i.e., no digits, no punctuations, etc). Describe a map-reduce design pattern to achieve the same.

CMI/02/05/2024/Final Exam May 2024

Chennai Mathematical Institute

Distributed Computing and Big Data

DURATION: 2 HOURS MAX MARKS: 35.

Instructions

- Please remember to mention your name and roll number in your answer sheet.
- This is an individual task. Do not discuss with anyone.
- This is a closed book exam. You are not allowed to carry books or cheatsheets.
- No electronic devices (calculators, laptops, etc) are allowed in the exam hall. Wherever heavy calculation is involved, you need not evaluate it to the final number unless it is explicitly asked for. For example, it is acceptable to leave the answer as $\frac{1}{1+\frac{5}{32}}$. You need not evaluate it to 0.865.
- First section has negative marks. No negative marks for the rest of the sections.

Section 1: All questions carry one mark each. -0.5 for wrong answers. Answer in True/False.

Question 1. One Petabyte space is enough to store 4 Million ebooks of size 1 MB each. True

Question 2. Based on Amdahl's law, we can expect a linear increase in speed-up for a specific job as we increase the number of processors. False

Question 3. Yet Another Resource Negotiator, is a resource management and job scheduling technology for the Hadoop distributed processing framework True

Question 4. One of the items in pig philosophy is that pigs live anywhere. True

Question 5. As per the principles of object-oriented programming, an object has an identity while a class does not. True

Question 6. An impedance mismatch occurs in relational databases when a relational database needs to be transformed into an object-oriented model. True

Question 7. BSON is a binary serialization format used to store documents in MongoDB. True

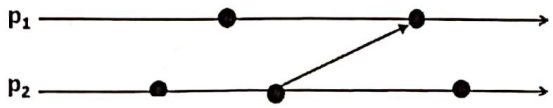
Question 8. NoSQL datastores are not ACID compliant. True

Question 9. Changing a block in a blockchain makes all the following blocks invalid. True

Question 10. A carrier hotel is a facility strategically based in a location closer to users that houses networks and cloud services. True

Section 2: All questions carry 2 marks each.

Consider the following space-time execution diagram while answering the questions in this Section.



Question 11. List all the happens-before relationships.

Question 12. Annotate the events using scalar time.

Question 13. Annotate the events using vector time.

Question 14. Annotate the events using matrix time.

Question 15. Identify an inconsistent cut.

Section 3: All questions carry 2 marks each.

Question 16. Consider the following Pig script.

```
Lines = LOAD 'file1' USING PigStorage() as (line:chararray);
Words = FOREACH Lines GENERATE FLATTEN(TOKENIZE(line)) AS word;
Groups = GROUP Words BY word;
Counts = FOREACH Groups GENERATE group, COUNT(Words) as Cnt;
Results = ORDER Counts BY Cnt ASC;
Dump Results;
```

Assume that the input file 'file1' contains the following two lines:

```
cmi is the best
the best in chennai is cmi
```

What does the pig script output?

Question 17. The following pig script was written to find the most expensive iphone. However, it has errors. Identify the errors and correct them.

Pig Script:

```
A = LOAD 'file2' USING PigStorage(',') AS (year:int,product:chararray,cost:int);
B = GROUP A BY $2;
C = FOREACH B GENERATE MIN(A.cost);
DUMP C;
```

Input File ('file2' contains year,product,cost):

```
2022, iphone, 50000
2023, iphone, 65000
2024, iphone, 72000
```

Expected output is 72000.

Question 18. How many nodes are created when the following three statements are executed by Neo4j?

```
1. CREATE (p:Person{name:'Venkatesh'})-[:Teaches]->(c:Course{name:'BigData'})
2. CREATE (p:Person {name:'Raj'})-[:StudentOf]->(o:Org{name:'CMI'})
3. MATCH (a:Person),(b:Org) WHERE a.name = 'Venkatesh' AND b.name = 'CMI'
   CREATE (a)-[:FacultyAt]->(b)
```

Section 4: All questions carry 3 marks each.

Question 19. Describe a map-reduce design for computing median of a large list of numbers. Assume that the input file contains 2 Million lines. Each line contains an integer ranging between 1 and 1000.

Question 20. Assume that Indian Railways wants to store the train running schedule (past and live status) information in MongoDB. Provide a database design along with at least one or two queries as example to indicate how you would query the data.

Question 21. Design a RESTful web service for a learning management system such as moodle. Include at least three items in your object model.

SAMANTA DEY (MDS202032)

DCBD Final Exam 2020-2022

Ans:1. (1) Here, $N=4$, $M=2$

Given: Atleast one muddy child.
The children don't know M's value.

Let d denote the set of children with mud [#d=2]
c denote " " " " without mud. [#c=2].

ROUND 1 :-

All of d and c say "No" since they don't know if they are muddy.
This is because, none of them knows the no. of muddy foreheads.

ROUND 2 :-

- Inference after round 1 : # muddy foreheads = >1. [This is because, if there was exactly one muddy forehead, the child with muddy forehead would see no other muddy forehead and could answer "Yes".
- The children in d set answer "Yes" [they know if they are muddy]
- The children in c answer "No" [they don't yet know if they are muddy]

Reason:- The children in d know that there are >1 muddy foreheads and each of the d children in d can see only other child in d's muddy forehead.

The children in c can see two muddy foreheads (of those in d) and cannot conclude anything about their own.

ROUND 3 :-

- Inference from Round 2 :- # muddy foreheads = 2. [since, both the children in d know they are muddy].
- The children in d answer "Yes"
- The children in c answer "Yes" [∵ all of them know whether each of them have mud on their own forehead]

Thus, after Round 3, all responses converge.

Ans:1 (2) $N=4$, $M>0$, $M=$ even

Notations and symbols: same as 1.(1)

- $M=2$

ROUND 1 :-

All of d and c say "No" since none of them

- $M=2$:-

Same as Ans:1 (1).

- $M=4$:-

Round 1 :-

All of d and c say "No" since all see 3 muddy foreheads.

Round 2 :-

- Inference from Round 1 :- # muddy foreheads >1 [same as 1.(1)].
- All of d and c say "No" since they are not yet able to figure out if they are muddy because all of them see 3 muddy foreheads and the total no. of foreheads with mud is 3 or 4.

Round 3 :-

- Inference from Round 2 :- # muddy foreheads >2 because o.w children in d would have said "yes".
- Still none of d and c can answer since they are not properly able to figure out if they themselves have mud on their foreheads. All say "No"

Round 4 :-

- Inference from Round 3 :- # muddy foreheads >3.
- All of d and c answer "Yes" since they are now sure that all of them have mud on their foreheads.

$\therefore$ Responses converge after Round 4.

Ans:2. Our RESTful web service will mainly do the following two tasks :—

- create a new registration for an user who had completed one dose of vaccination or has not taken any dose of vaccination
- return the vaccine information for the users who have administered a single dose of vaccine [information about the type/vial].

Two resources :-

- ~~Name~~ Person
- ~~Conta~~ Vaccine

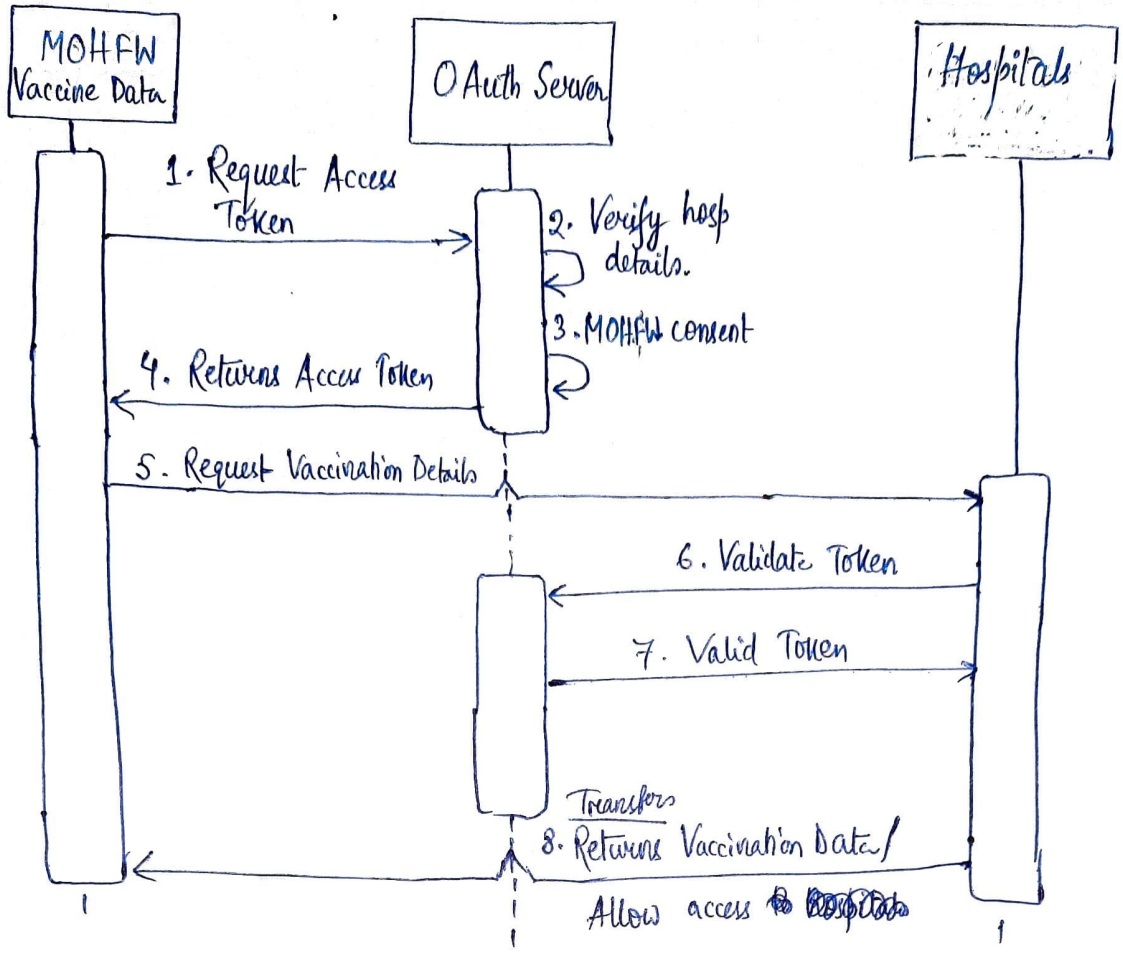
Design of the RESTful API :—

- Object Model :—
  - Person, Vaccine
- Creating Model URIs :—
  - /person/{pname}
  - /person/{pname}/vaccine
  - /person/{p-name}/vaccine/{vaccine-name}
- Determining representations :—
  - Represent all person's information as an XML/JSON
  - Represent all vaccine names as an XML/JSON.
- Assigning HTTP Methods :—
  - create registration for users who are new or completed one dose → HTTP POST
  - return information about type of vaccine taken → HTTP GET

Here, 'POST' is non-idempotent

'GET' is idempotent.

Ans 3.



When a particular hospital needs to ~~access~~ transfer the resources on vaccination data, through a secure method, the OAuth Server asks the hospitals to login for a secure transaction/transfer of data and after verifying the hospital details, when the ~~Hospital~~ MOHFW requests access to the vaccine data, the hospitals validates the token and returns the data and allows access to the data for ~~the Hospitals~~ MOHFW.

This has been represented using a sequence diagram as above.

Ans: 4. Class Diagram for MOHFW scenario :—

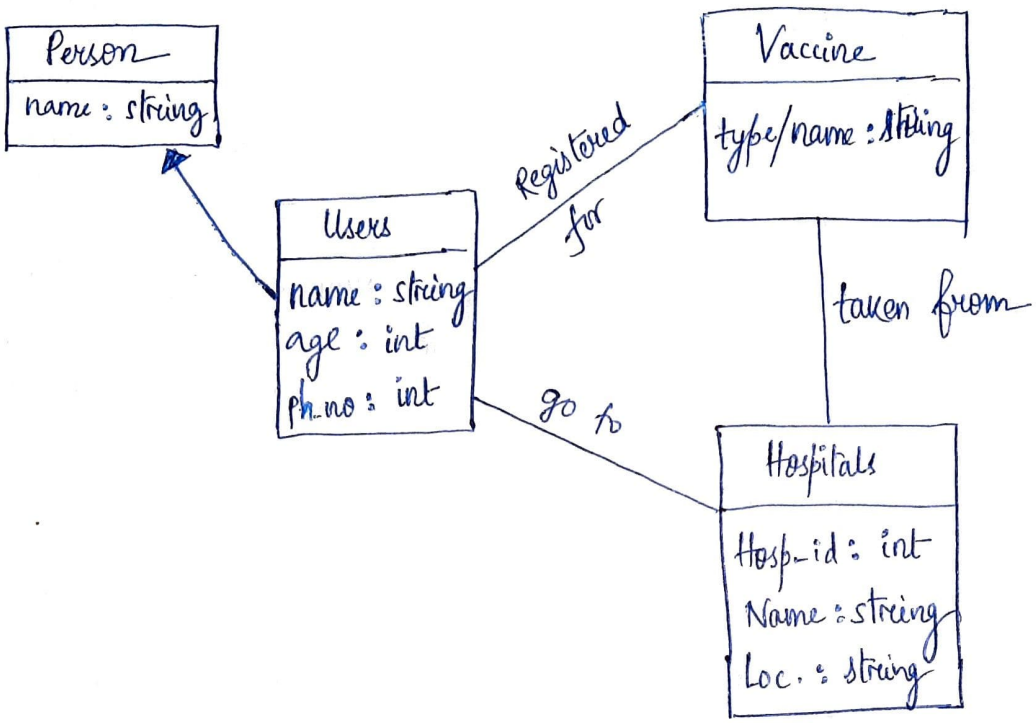


Fig: Class Diagram to demonstrate vaccination situation

Here, the classes are :—

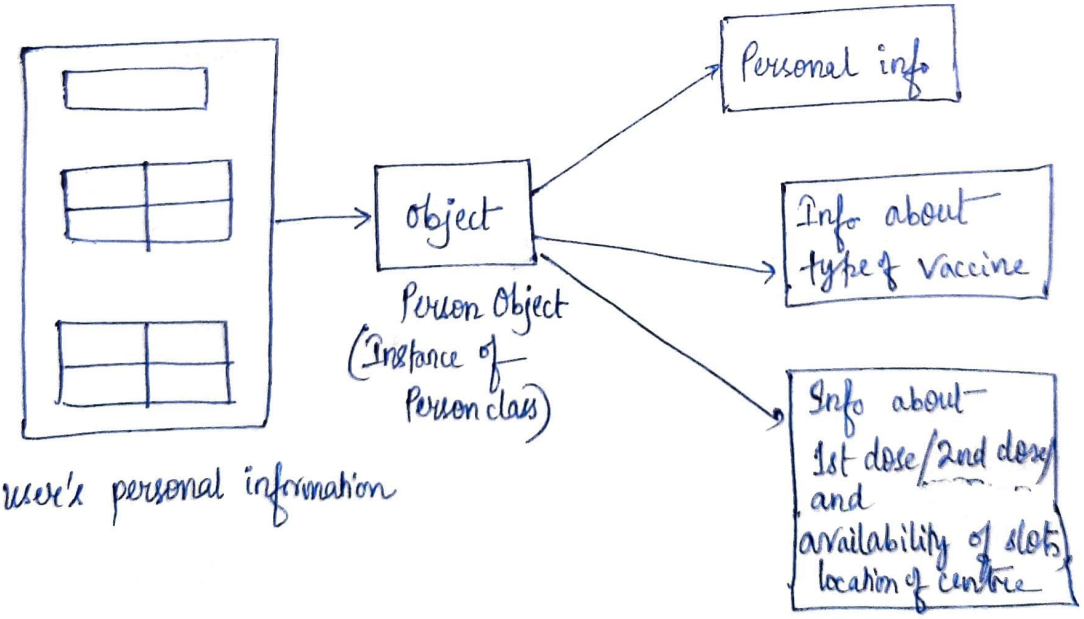
- Users
- Vaccine
- Hospitals

- Users — Person is the "IS-A" Relationship.

ORM can be used in the following ways in this situation :—

- For transfer of vaccine data from hospitals to MOFHW, the hospitals could simply provide its details and the data to MOFHW and the ORM abstracts details like verification through access tokens, generating requests and validating them etc.

- For users' registration at the MOFHW portal for vaccination, the users just enter credentials or generate request for slot details, vaccine availability and the system provides the details to the user and abstracts all the intermediate steps like separating the user credentials to separate tables, fetching data related to the logged in user from the server etc. from the user.



A user's personal information

Ans: 7.

(1) Here, we want to choose two processes as co-ordinators from a group of n independent processes.

We assume that every active process in the system has a priority no. associated and the processes can send messages to all other processes in the system.

We want to elect the top two processes with the highest priority nos as the co-ordinators.

If any host thinks that the co-ordinator has failed, it tries to elect itself by sending a message to the highest numbered processors. If any of them answers, the host loses and each of the processors will call election and try to win themselves.

If none of the highest priority no. co-ordinators answer, then the host becomes the co-ordinator.

If a new processor arrives, it again calls for an election.

~~This continues two times~~

After getting the 1st co-ordinator, the above process repeats itself excluding the new co-ordinator to elect the next co-ordinator.

(2) Similarly, for ~~n~~ electing m processes, continue the procedure m times keeping in mind whether new processes arrive and in that case, recall the election.

Ans:6. Let us consider a vector clock consisting n elements $V = (v_1, \ldots, v_n)$ which can be encoded by distinct prime nos. $p_1, p_2, \ldots, p_n$.

~~A method could be to encode~~

Here, actually, we have encoded the entire vector V to a unique number using $p_1, p_2 \ldots, p_n$.

The encoded representation of $V = p_1^{v_1} \cdot p_2^{v_2} \cdots p_n^{v_n}$

This works well because the above product is distinct for any permutation of $\{v_1, v_2, \ldots, v_n\}$

This encoding of n-entries to a single no. reduces the space needed to represent vector clocks.

AVIRUP DAS - MDS202013

DCBD - ENDSEM

07/07/2021

1) Let us consider the following two resources :-

i) Person ii) Vaccination Record

Here we could identify the vaccine receiver from a unique ID. Now when this person receives his first dose of vaccine, we create a new record for that patient using the POST command (non-idempotent method). When this person is about to receive the second dose, we need to check the type of vaccine, no. date of first dose, etc. so the older records we could use GET method (idempotent). Finally, we will update the second dose details using PUT (idempotent).

- Object model : Person, Vaccination Record
- Model URIs : /Person/{unique ID}
  /Person/{unique ID}/Record
  /Person/{unique ID}/Record/{first vaccine date}
  /Person/{unique ID}/Record/{second vaccine date}
  /Person/{unique ID}/Record/{Vaccine type}

We will alternate both the Person and Record details as JSON or XML.

To create record:
HTTP POST /Person/{unique ID}/Record/{Vaccine type}

To update record: HTTP PUT /Person/{unique ID}/Record/{first vaccine date}
HTTP PUT /Person/{unique ID}/Record/{second vaccine date}

To retrieve record: HTTP GET /Person/{unique ID}/Record/

7) i) Let us assume that the system already has two coordinator processes. Now if some process P sends a message to the coordinators, there will be a few cases:
i) No coordinators respond, ii) Both respond, iii) One of them responds.

If both coordinators respond then we are alone.
If one/or few respond then we choose new coordinators.

→ Process P will send an election message to all the processes which have some high priority scores.

→ P will wait for a time slot for it to receive response. Now if it does not receive any messages then it chooses itself as a coordinator and sends another message for the second coordinator, and wait for time slot and so on.

→ If P receives some messages, it will select those two processes as coordinators which have the highest priority score amongst the respondents.

→ If P receives one message, it will select that process as coordinator and send a second message. If there is no response then it elects itself.

Note that if at any stage P receives a message from process, then it will send a reconfirmation message and wait for time slot t, to for confirmation. If the confirmation is received, the processes are confirmed as coordinators, otherwise the whole process starts again.

7)(2) We will take a remote approach for $n (< 2)$ processes. First a message is sent to the coordinators of all of them respond then we are done, if atleast one of them do not respond then we have to elect that many coordinators. If none of them respond —

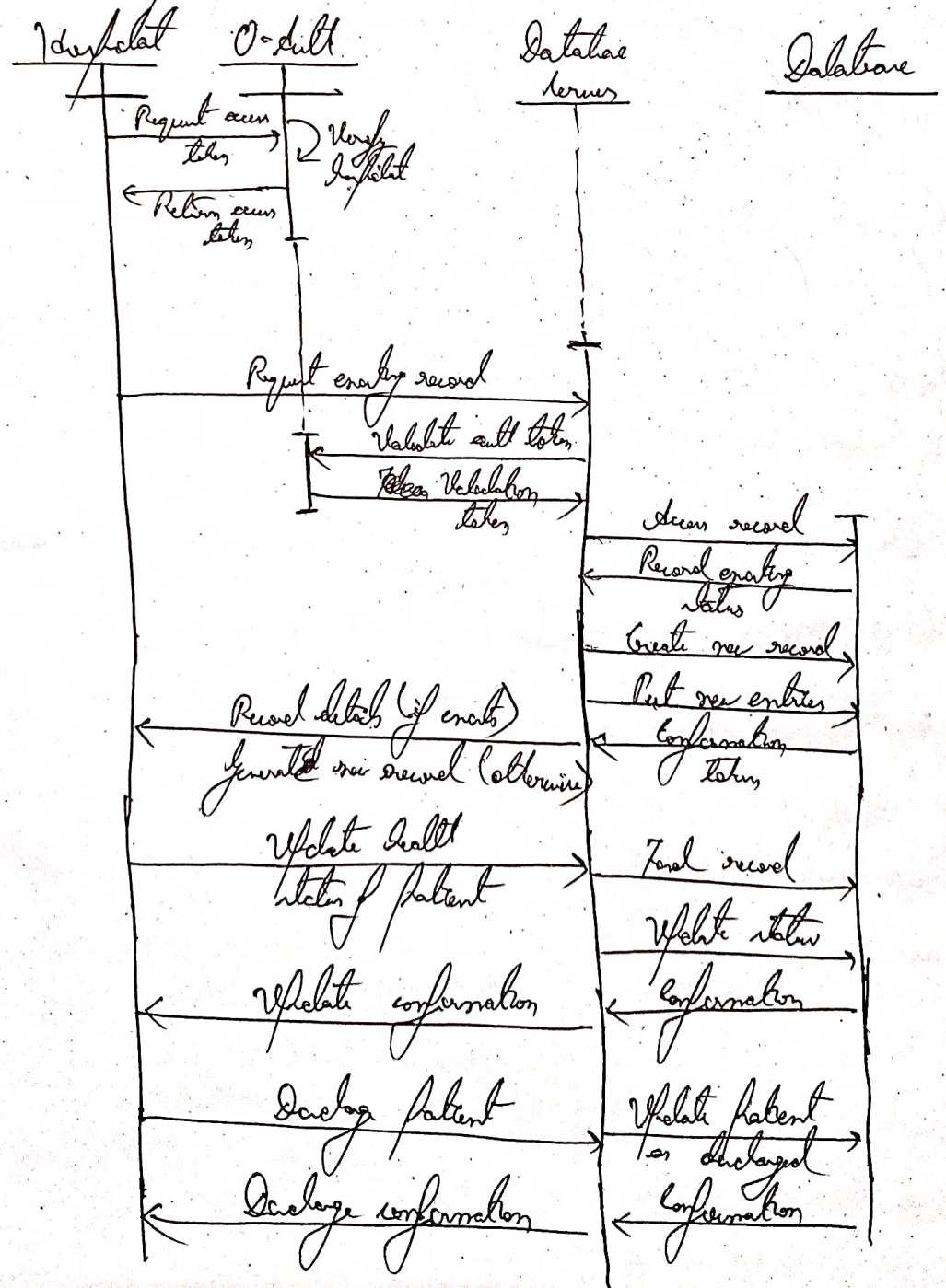
→ P sends a message to all processes for election and waits for time period t. If there is no response then it elect itself as one of the coordinators and moves forward with the election. If i responses are there (i = 1, 2, ... n−1) then it will elect the i-processes and itself.

→ If there are more than n responses, it will select those processes which have the highest priority score.

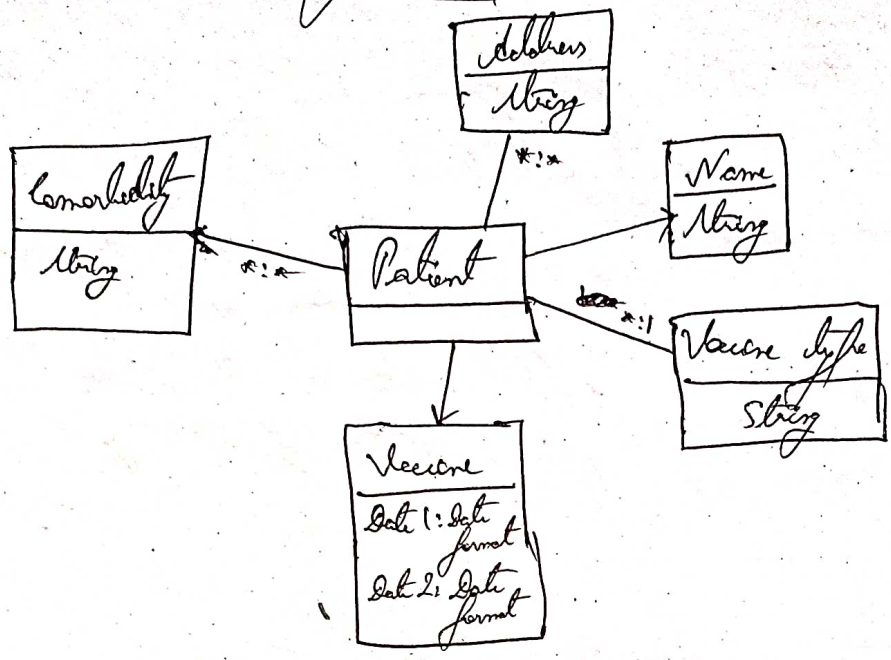
→ In this first case of message passing, the process P will wait for time t for the processes to respond otherwise it will elect itself and move on & will the election procedure for the rest of the coordinates remaining.

→ After the first case of message passing, process P will ask for a confirmation message from the 4 coordinates and wait for time t, for the coordinates to respond, if it receives confirmation messages within t, then it moves forward with the procedure otherwise we start again from the beginning.

3) First the hospital needs to send a message to the oAuth server so that oAuth server can authenticate the hospital as a legit account holder and pass on the request to the resource server. After verification, the hospital now has access to the resource server which has the database. Now the Thus hospital first tries to find the existing database or, if found then will be a request to update the same. If the database is non existent, the server will create a new one and will send the details back by the hospital.

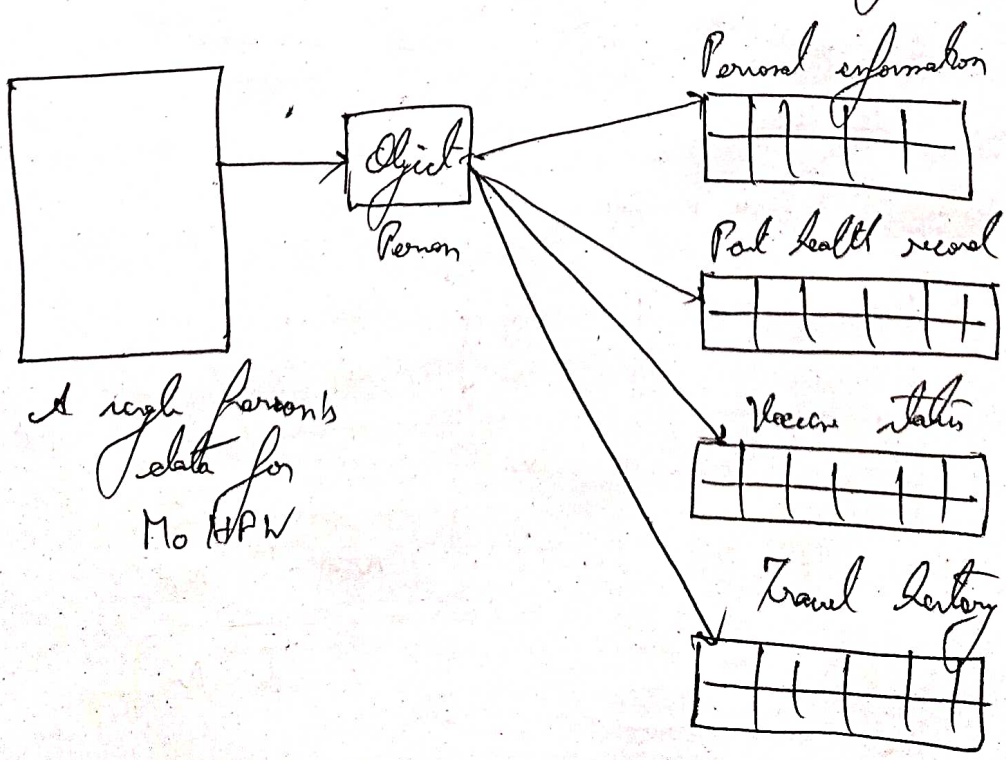


4) Partial class diagram :-



ORM :-

We have seen that it is much more intuitive and easy to represent data and in terms of object while being able to write simple queries. Clearly, there could be different types of object for the patient, for example - Personal information, Past health record, Vaccine status, Travel record. This can be represented using ORM.



1) (1) If N=4, M=2

1st round :-

All of the children will say no since they are not aware of the exact count. Although they will see the other children's heads, the exact count will be unknown since there is no info about their own status. So the a given child will either see 2 muddy heads or 1 muddy head depending on whether he himself is a muddy child.

2nd round :-

All of them conclude that atleast 2 of them are muddy. The two muddy children will see 1 muddy head and the clean headed children will see 2 muddy heads. Their clean headed won't be able to conclude about their own status, while the muddy ones will conclude that they have mud on their foreheads.

3rd round :-

From the second round they concluded that 2 children are muddy and they can identified themselves as muddy, this couldn't not have been possible if there were more. So from the previous round's answers, the clean children will identify themselves as clean in this round.

Round \ Child	1	2	3	4
1	NA	NA	NA	NA
2	Y	Y	NA	NA
3	Y	Y	N	N

NA → not sure

1)(2) If $N=4$ and M is a positive even number, M can be 2 or 4.

1)(1) → $M=2$ case has been discussed. Let us discuss it for $M=4$.

1st round → All of the children have mud on their foreheads and all see 3 other muddy children and no conclusion about themselves. They conclude that there are atleast 3 muddy children. But they can't pass in this manner, so there are 2, 3 or 4 muddy children.

2nd round → They concluded that there are atleast 3 muddy children but they still can see 3 muddy each and are not sure if they have mud on their own forehead. So, the ~~2nd round~~ ~~In the 2nd round~~ central inference would be there are atleast 3 muddy children.

3rd & round → In this round they know that 3 are muddy and atleast 3 should be muddy, so there is no strong conclusion.

4th round → In the final round everyone says yes as if exactly 3 were muddy then it could have been identified from the previous round - In the previous round if they would have seen 2 muddy and 1 non-muddy then they would have concluded in that round itself. So there are 4 muddy children and here all of them say yes.

Round \ child	1	2	3	4	
1	NA	NA	NA	NA	→ All muddy
2	NA	NA	NA	NA	
3	NA	NA	NA	NA	
4	4	4	4	4	

6) Note that when we are using vector clocks, it is difficult to determine the optimum number of bits that will be used to implement the integer values of the clock. If this number is very small, there might be a risk of overflow if there are millions of events. If the number is too large, there is an additional cost of storage and maintenance. Hence large clocks should is undesirable. Thus we can have to take a rough estimate of the no. of bits and continue the process. If we are in a situation when there is a risk of overflow, we can reset the vector clock for that particular process. Resetting the clocks helps us retain the functionality of the vector clock. We can even prevent clock overflow by carefully selecting the condition under which the clock is reset.

---

5) Let us take three persons.

Next we plot this matrix row-wise, meaning the row gives us the states of that person's branch. Now we plot this row into cells and pass through a mapper where the key is the person's name (row corresponding row) and the value is 1 if that cell has 1 or 0. These outputs are then shuffled and passed through a reducer where the values are reduced. Then we get the final output where the highest value has the highest no. of branch.

Input

Person	Branch visited		
	X	Y	Z
X	0	1	1
Y	0	0	1
Z	0	1	0

Shuffle

X	0	1	1
Y	0	0	1
Z	0	1	0

Reduce

X	2
Y	1
Z	1

→ X has the highest no. of branch