

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

REPLICATED DICTIONARY AND LOG

Replicated dictionary problem

- Efficient solutions to the replicated log and dictionary problems. Wu and Bernstein
PODC 84
- Replication is a fundamental method for fault-tolerance. <https://powcoder.com>
- Replication is also used for performance. Add WeChat powcoder
- Dictionary is a common data structure which is often fully replicated on all sites.
- (Blockchains are also fully replicated)

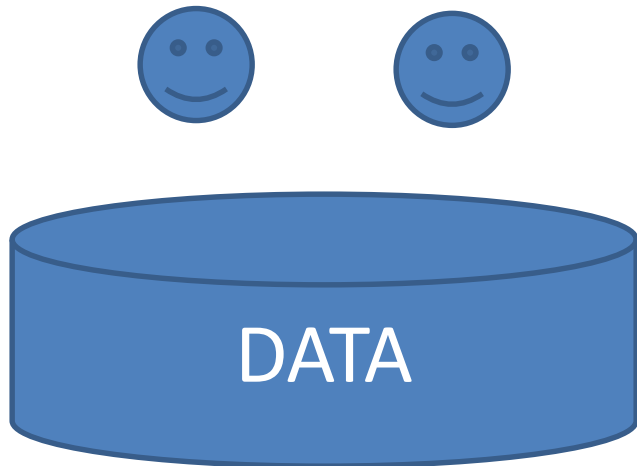
Replication motivation

- What if data is stored in **ONE place**?

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



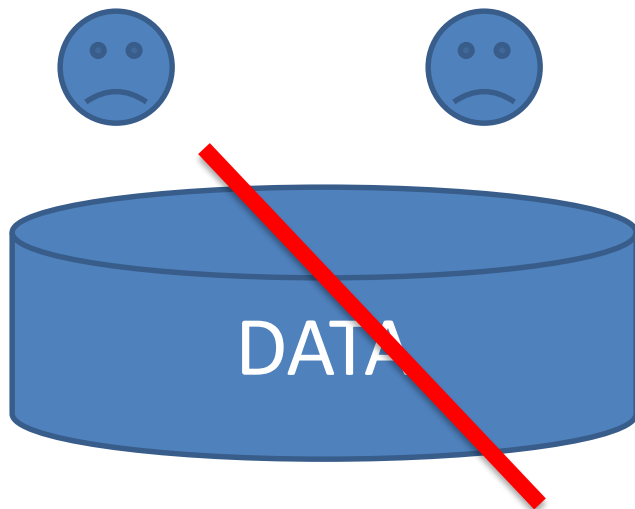
Replication motivation

- What if data is stored in **ONE place**?
 - **Failures** cause data to be **inaccessible** (single point of failure)

Assignment Project Exam Help

<https://powcoder.com>

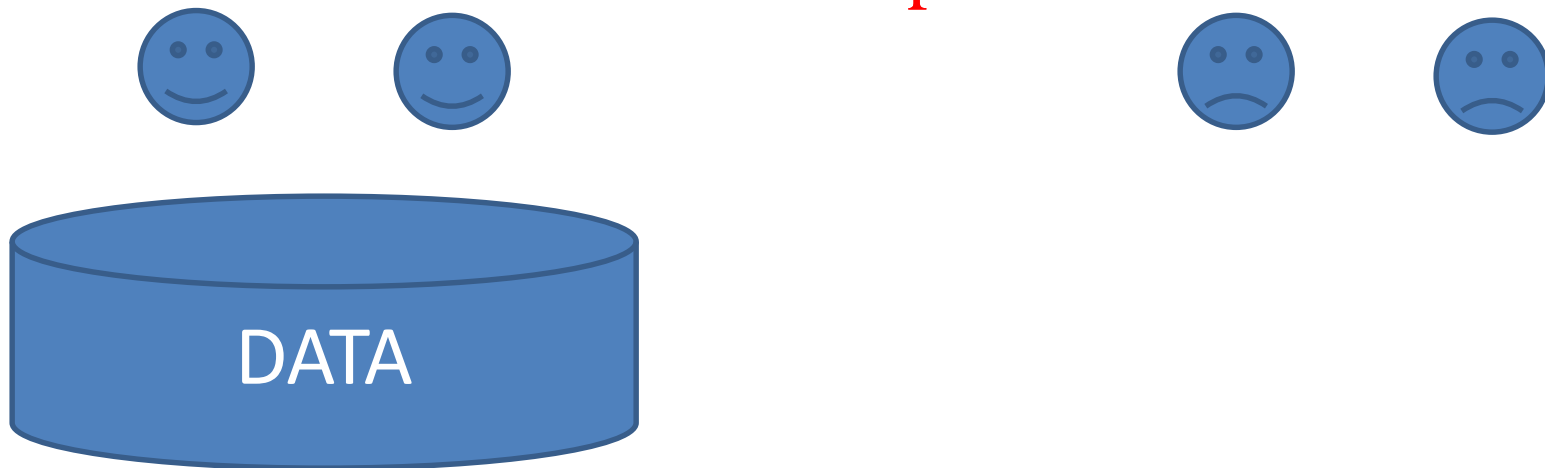
Add WeChat powcoder



CRASH!!

Replication motivation

- What if data is stored in **ONE place**?
 - Failures cause data to be **inaccessible** (single point of failure)
 - Users far away experience **large latency** to access data



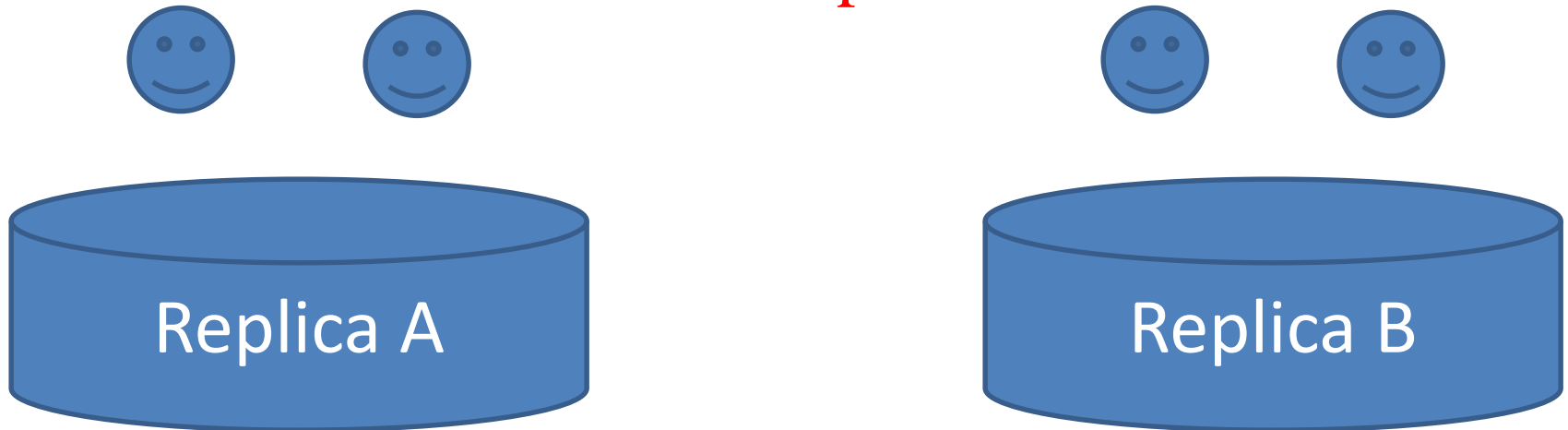
Replication

- Replication requires maintaining data in multiple locations

– Identical copies?

<https://powcoder.com>

Add WeChat powcoder



Basic assumptions

- **Very general**: Sites may crash, links may fail, partitioning.
- **Asynchronous** System
- Each site maintains a **local clock** (a counter).
- Use **Lamport's event execution model** and happens-before relation.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Assignment Project Exam Help
Motivating example for causality

<https://powcoder.com>

Add WeChat powcoder
Micro-blogging application

Causality in micro-blogging

- A micro-blogging website has users and a micro-blog
- One global stream of users blog
- In what order should the stream display blogs?
- In a non-replicated deployment, order by time of arrival.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

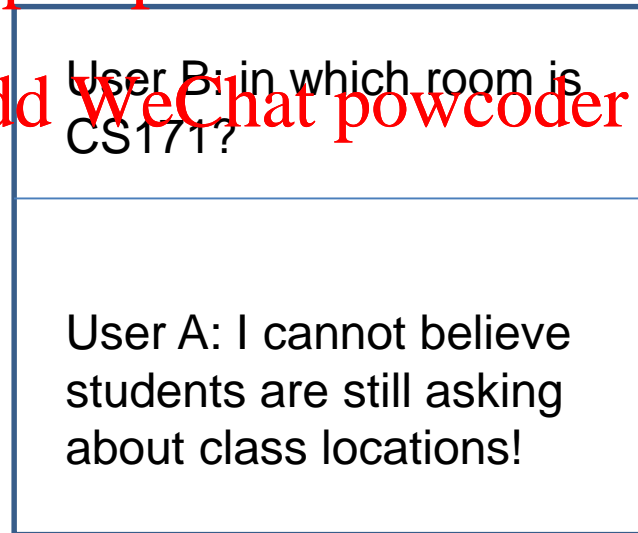
Causality between blogs

- A user A blogs blog a1 as a consequence of an earlier blog b2 (or blogs) by another user (users) B.

Assignment Project Exam Help

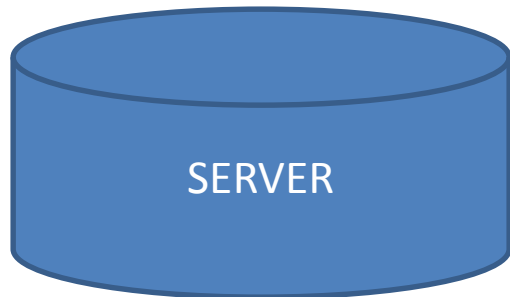
<https://powcoder.com>

Add WeChat powcoder



Causality between blogs

- This causality is trivially captured in non-replicated systems. User A made a comment after seeing what caused it (blog by User B) in the stream, which means it already exists.



Add WeChat powcoder

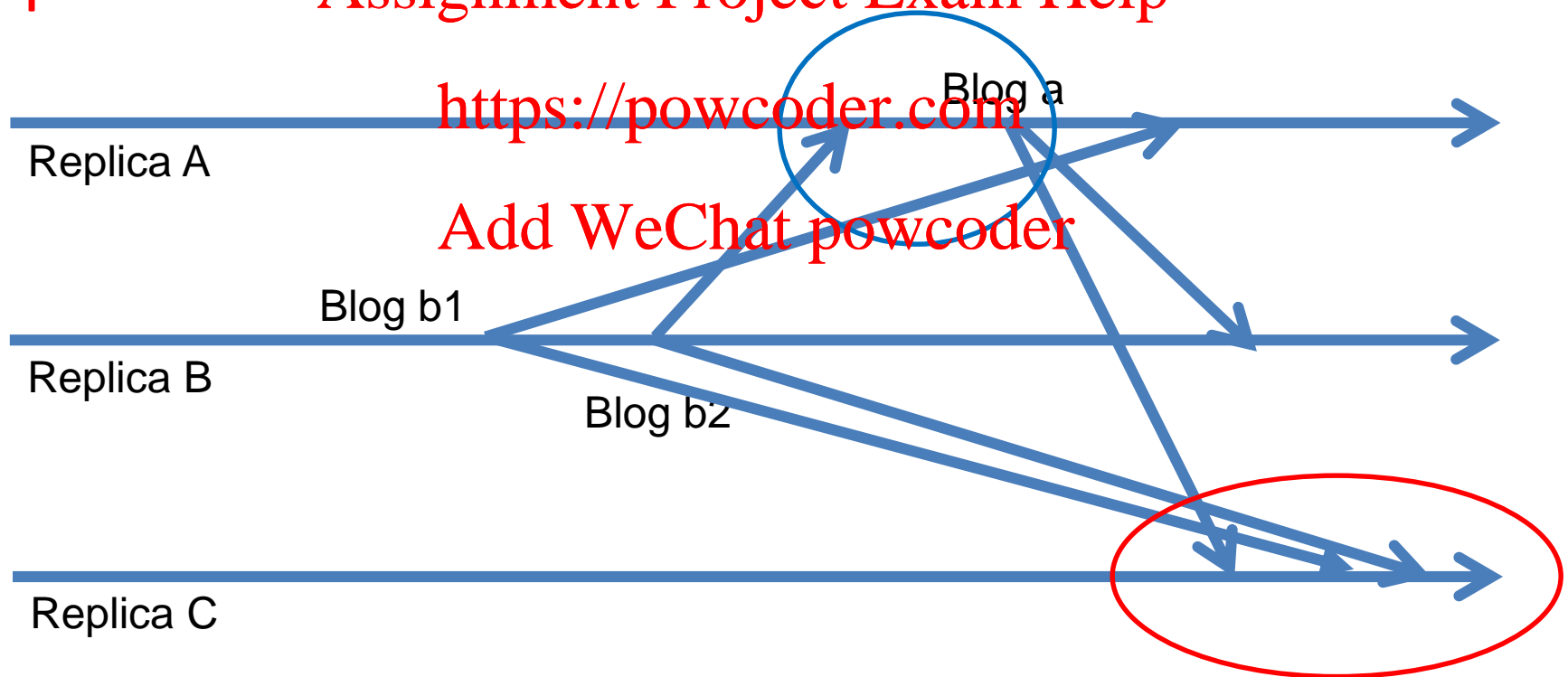


Causality between blogs

- For three replicas, A, B, and C. If blogs are just replicated asynchronously it will not guarantee causality.
- Naïve-replication:
 - When a blog is posted send to all replicas.
 - When a replica receives a blog, append to log.

Causality between blogs

- Blog a is caused by blog b2,
- at replica C blog a is ordered before blog b2;
- **problem!** Assignment Project Exam Help



Log stores

- An append operation inserts a record at the head of the log

Assignment Project Exam Help

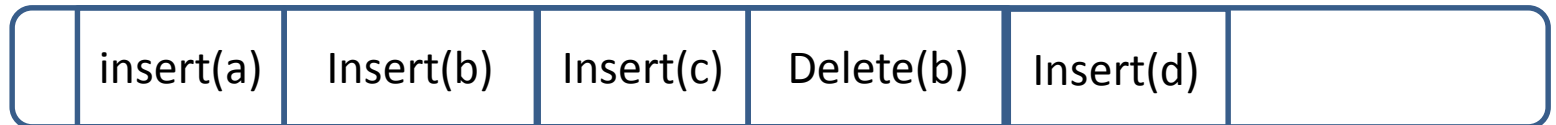
- Changing the content of a record: delete record and then insert

<https://powcoder.com>

- Reading log => all log content.

Add WeChat powcoder

Head of log



The Log- Example

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Site 1

--	--	--	--	--	--

Log

The Log- Example

- $I(x)$

Assignment Project Exam Help

Site 1

<https://powcoder.com>

$I(x)$	Add	WeChat	powcoder
--------	-----	--------	----------

Log

The Log- Example

- $I(y)$

Assignment Project Exam Help

Site 1

<https://powcoder.com>

$I(x)$	$I(y)$				
--------	--------	--	--	--	--

Add WeChat powcoder

Log

The Log- Example

- $D(y)$

Assignment Project Exam Help

Site 1

<https://powcoder.com>

$I(x)$	$I(y)$	$D(y)$			
--------	--------	--------	--	--	--

Add WeChat powcoder

Log

The Log- Example

- $D(x)$

Assignment Project Exam Help

Site 1

<https://powcoder.com>

$I(x)$	$I(y)$	$D(y)$	$D(x)$
--------	--------	--------	--------

Add WeChat powcoder

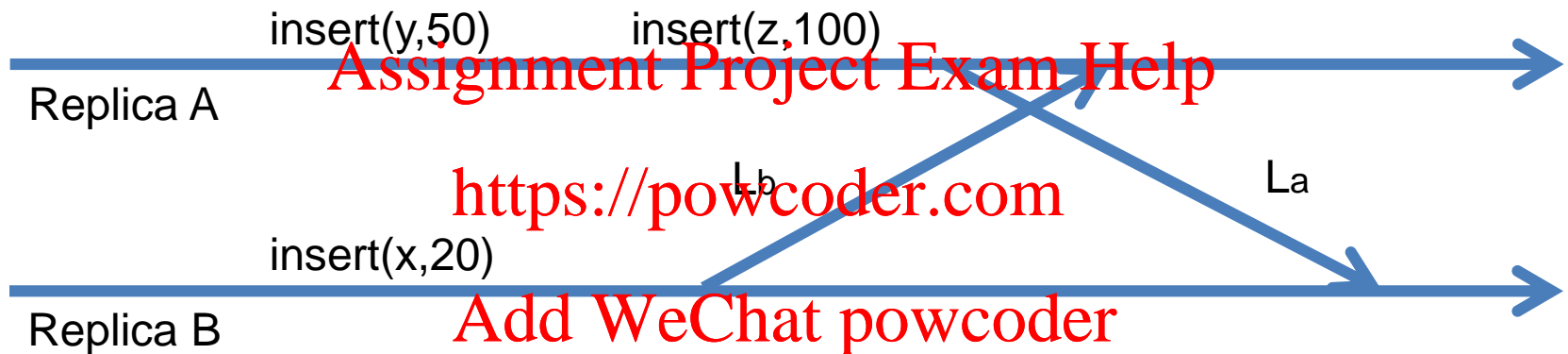
Log

The log problem

- Each **site** maintains a **copy of the log**.
- The **log contains local events**, i.e.,
 - insert **Assignment Project Exam Help**
 - delete
- The goal of the algorithm is to **keep all copies of the log up to date**.
<https://powcoder.com>
Add WeChat powcoder
- L_i is the copy of the log at site i .
- $L(e)$ is the contents of log $L_{\text{node}(e)}$ **immediately after event e is executed**.

An example of a log

- Logs are transmitted between replicas



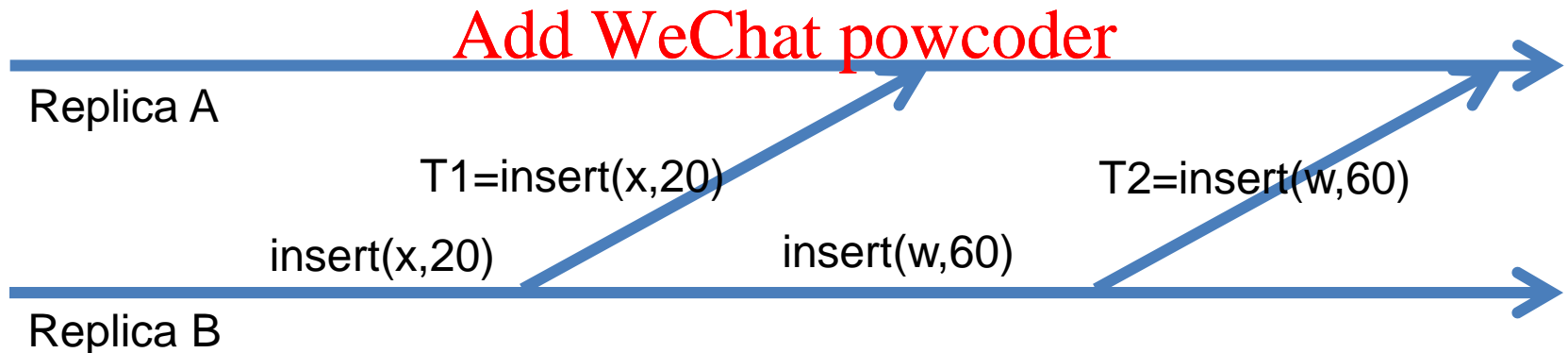
- What to include in the transmitted log?
- $L_b = \{\text{insert}(x,20)\}$
- $L_a = \{\text{insert}(y,50), \text{insert}(z,100)\}$

The log problem

- Log Problem: find an algorithm that maintains the log such that given an execution $\langle E, \rightarrow \rangle$,
 $\forall \text{ events } e, f: \text{ if } f \rightarrow e \text{ then } f \text{ is in } L(e)$
- General approach:
 - For each local event, $\text{Add WeChat powcoder}$ in the local log.
 - Exchange logs to update other sites
- Main question: when to exchange logs? With application communication to capture the happens before relation.

Attempt at Solution 1

- A solution:
 - Site i **sends** to site j all records in the log that were inserted since i last sent a message
 - **WHY INCORRECT?**
 - Log message lost – log messages reordered



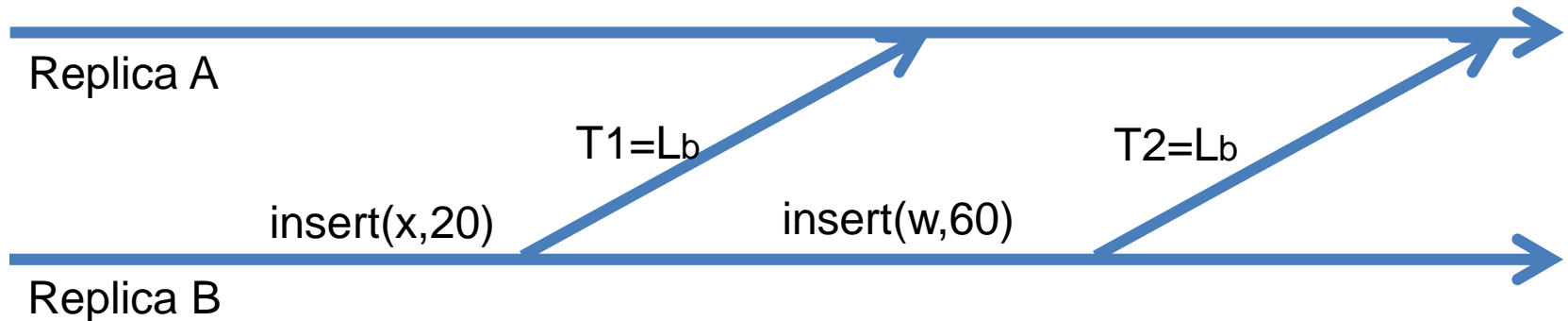
Attempt at Solution 2

- Another solution:

- each site i includes its $\log L_i$ with each message.
- On receiving a message, a site j incorporates all new event records.

– **BAD:** <https://powcoder.com>

- Entire log sent with each message
- Entire log kept at each node.



Efficient solution for log problem

- Observation 1: Once i “knows” that j “knows” of an event e (which may have occurred on site k), then i does not need to include event e in messages to j .
Assignment Project Exam Help
<https://powcoder.com>
- Observation 2: Once i “knows” that **all** sites “know” about an event e , then i does not need to keep a record of e in its local log.
Add WeChat powcoder

2 Dimensional Time-Table

- $TT_i[n,n]$:

k

	Assignment Project Exam Help https://powcoder.com	
j	Add WeChat t powcoder	

- if $TT_i[j,k] = t$,
then site i knows that site j has learned of all events that occurred at site k up to time t .

The 2 dimensional timetable

- site j might actually **know about more events**, but site i may not be aware of it.

Assignment Project Exam Help

- $TT_i[i,i]$ is the **value of clock at site i** .
<http://powcoder.com>

Add WeChat powcoder

- $TT_i[i,k]$ is the value of clock at site k of the most recent event at site k that site i is aware of.

Two dimensional timetable

- Let $\text{hasrec}(\text{TT}_i, e, k)$ be true iff

$$\text{TT}[k, \text{node}(e)] > \text{time}(e)$$

<https://powcoder.com>

- The algorithm must guarantee that if $\text{hasrec}(\text{TT}_i, e, k)$ is true, then site k has learned of event e .
- Note: site i need **not send** a record of event e to site k if $\text{hasrec}(\text{TT}_i, e, k)$ is true.

The Replicated Log- Example

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

TimeTable

0	0	0
0	0	0
0	0	0

Site 1 _____



Log

Site 1 _____

Log

--	--	--	--

TimeTable

0	0	0
0	0	0
0	0	0

Assignment Project Exam Help

Site 2 _____

Log

--	--	--	--

<https://powcoder.com>

TimeTable

0	0	0
0	0	0
0	0	0

Add WeChat powcoder

Site 3 _____

Log

--	--	--	--

TimeTable

0	0	0
0	0	0
0	0	0



Log

I(X)			
------	--	--	--

TimeTable

1	0	0
0	0	0
0	0	0

Assignment Project Exam Help



Log

--	--	--	--

TimeTable

0	0	0
0	0	0
0	0	0



Log

--	--	--	--

TimeTable

0	0	0
0	0	0
0	0	0

<https://powcoder.com>

Add WeChat powcoder



Log

I(X)	I(Y)		
------	------	--	--

TimeTable

2	0	0
0	0	0
0	0	0

Assignment Project Exam Help



Log

--	--	--	--

TimeTable

0	0	0
0	0	0
0	0	0



Log

--	--	--	--

TimeTable

0	0	0
0	0	0
0	0	0

<https://powcoder.com>

Add WeChat powcoder



Log

I(X)	I(Y)		
------	------	--	--

TimeTable

2	0	0
0	0	0
0	0	0

Assignment Project Exam Help



Log

I(Z)			
------	--	--	--

TimeTable

0	0	0
0	1	0
0	0	0

<https://powcoder.com>

Add WeChat powcoder



Log

--	--	--	--

TimeTable

0	0	0
0	0	0
0	0	0



Log

$I(X)$	$I(Y)$	$D(Y)$	
--------	--------	--------	--

TimeTable

3	0	0
0	0	0
0	0	0



Log

$I(Z)$			
--------	--	--	--

TimeTable

0	0	0
0	1	0
0	0	0



Log

--	--	--	--

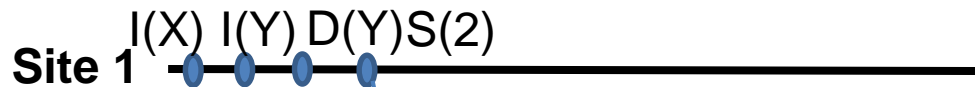
TimeTable

0	0	0
0	0	0
0	0	0

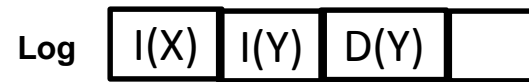
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Site 1 knows that site 2 knows till clock 0 about its local events



TimeTable

3	0	0
0	0	0
0	0	0



Site 1 sends $[(I(X), 1), (I(Y), 2), (D(Y), 3)]$ and its timetable



TimeTable

0	0	0
0	1	0
0	0	0

Site one send all its log local events with clock > 0



TimeTable

0	0	0
0	0	0
0	0	0

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

How to update the timetable?

3	0	0
0	0	0
0	0	0

 +

0	0	0
0	1	0
0	0	0

 =

0	0	0
3	1	0
0	0	0

Assignment Project Exam Help
<https://powcoder.com>

- Add WeChat powcoder
- Max of all elements
 - Max of times in local i'th row and remote k'th row.

Log maintenance

- Initialize all entries in TT to 0.
- For each local operation, **insert** a copy in the **local log**.
- With each send operation from site i to site k : **piggyback** **TT** + the following subset of the local log L_i : all records e such that **hasrec(TT, e) is not true**.
- On receipt of a message from site k by site i :
 - **incorporate all new events** into local log
 - **update TT**:
 - **Max** of all elements
 - **Max** of times in **local i 'th row** and **remote k 'th row**.

Dictionary problem

- Assume we want to maintain a *replicated dictionary* with *insert, delete and lookup*.
- On receipt of a message with partial log & TT:
 - Update local copy of the dictionary
 - Update local copy of TT as before
 - Garbage collect local log from any records that correspond to events e such that
 - \exists site j such that $\text{hasrec}(\text{TT}_j, e, j)$ is not true

Replicated Dictionary- Example

Assignment Project Exam Help

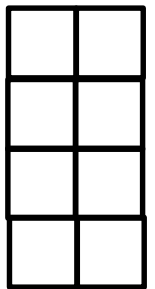
<https://powcoder.com>

Add WeChat powcoder

Site 1 _____



Log



Dictionary

Replicated Dictionary- Example

- $I(x)$

Assignment Project Exam Help

Site 1

<https://powcoder.com>

$I(x)$ Add WeChat powcoder

Log

x	...

Dictionary

Replicated Dictionary- Example

- $I(y)$

Assignment Project Exam Help

Site 1

<https://powcoder.com>

$I(x)$	$I(y)$				
--------	--------	--	--	--	--

Log

X	...
Y	...

Dictionary

Replicated Dictionary- Example

- $D(y)$

Assignment Project Exam Help

Site 1

<https://powcoder.com>

$I(x)$	$I(y)$	$D(x)$	$D(y)$		
--------	--------	--------	--------	--	--

Log

X	...

Dictionary

Replicated Dictionary- Example

- $D(x)$

Assignment Project Exam Help

Site 1

<https://powcoder.com>

I(x)	I(y)	D(y)	D(x)
------	------	------	------

Log

x	...

Dictionary

Add WeChat powcoder

Site 1 _____

Log

--	--	--	--

TimeTable

0	0	0
0	0	0
0	0	0

Dictionary

Assignment Project Exam Help

Site 2 _____

Log

--	--	--	--

TimeTable

0	0	0
0	0	0
0	0	0

Dictionary

<https://powcoder.com>

Add WeChat powcoder

Site 3 _____

Log

--	--	--	--

TimeTable

0	0	0
0	0	0
0	0	0

Dictionary



Log

I(X)			
------	--	--	--

TimeTable

1	0	0
0	0	0
0	0	0

Dictionary

X	...



Log

--	--	--	--

TimeTable

0	0	0
0	0	0
0	0	0

Dictionary



Log

--	--	--	--

TimeTable

0	0	0
0	0	0
0	0	0

Dictionary

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Log

I(X)	I(Y)		
------	------	--	--

TimeTable

2	0	0
0	0	0
0	0	0

Dictionary

X	...
Y	...

Assignment Project Exam Help



Log

--	--	--	--

TimeTable

0	0	0
0	0	0
0	0	0

Dictionary

<https://powcoder.com>

Add WeChat powcoder



Log

--	--	--	--

TimeTable

0	0	0
0	0	0
0	0	0

Dictionary



Log

I(X)	I(Y)		
------	------	--	--

TimeTable

2	0	0
0	0	0
0	0	0

Dictionary

X	...
Y	...



Log

I(Z)			
------	--	--	--

TimeTable

0	0	0
0	1	0
0	0	0

Dictionary

Z	...



Log

--	--	--	--

TimeTable

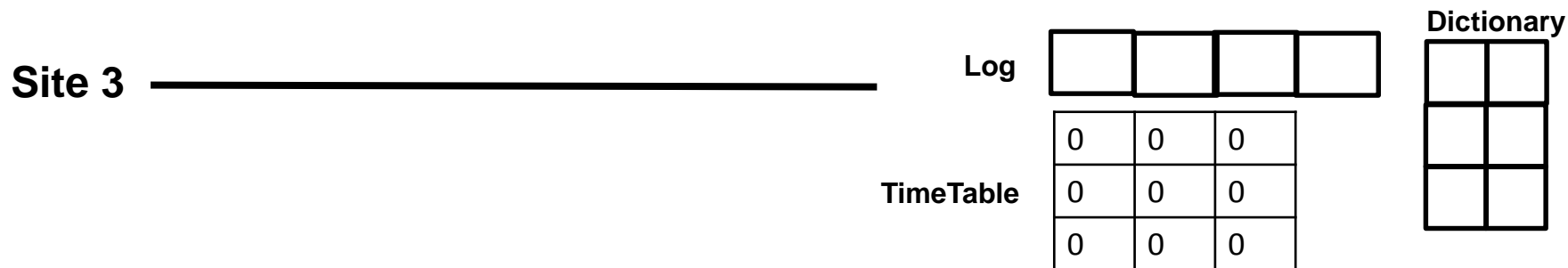
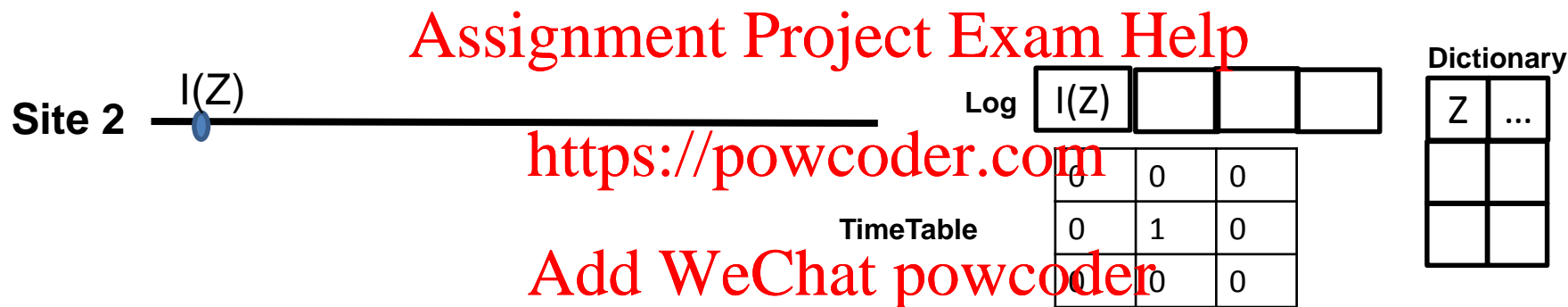
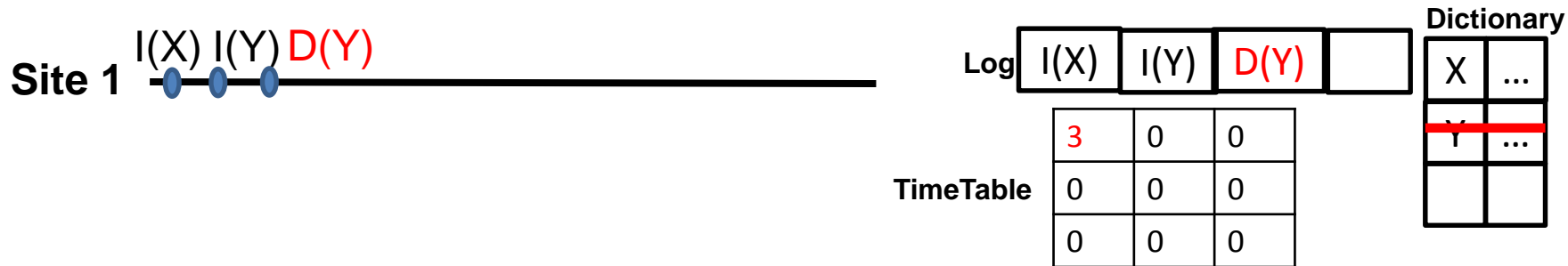
0	0	0
0	0	0
0	0	0

Dictionary

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder