Designing A Low-Latency Cuckoo Hash Table for Write-Intensive Workloads Using RDMA

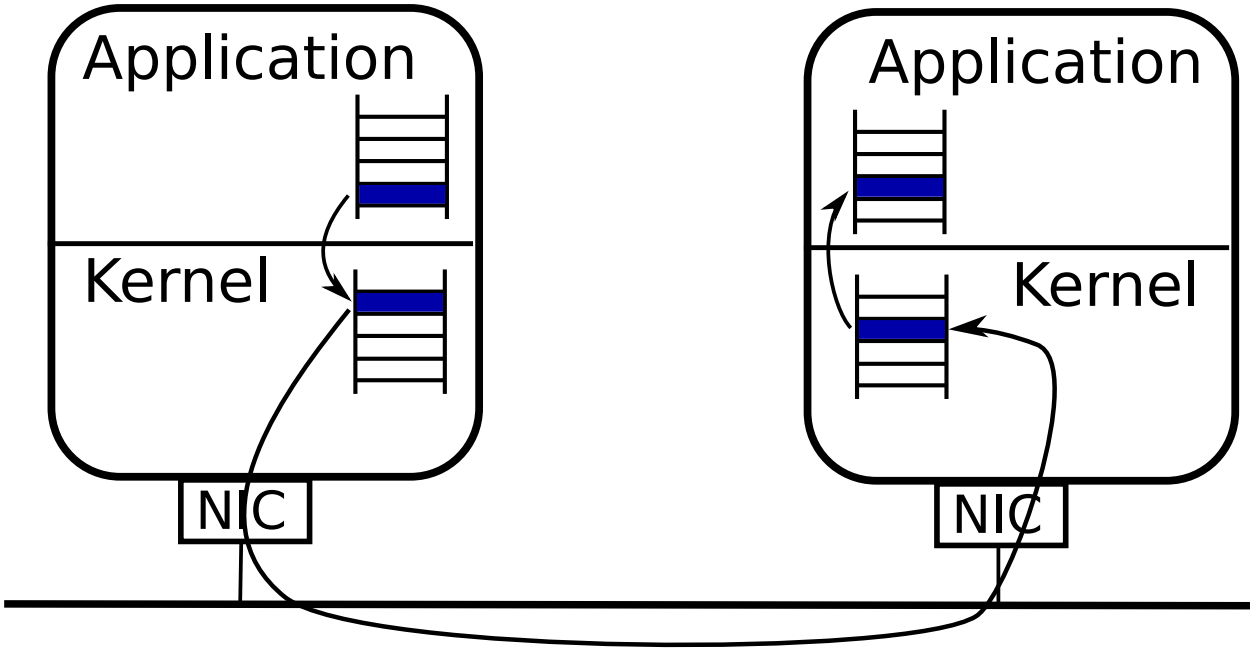
Tyler Szepesi, Bernard Wong, Ben Cassell, Tim Brecht

Cheriton School of Computer Science
University of Waterloo

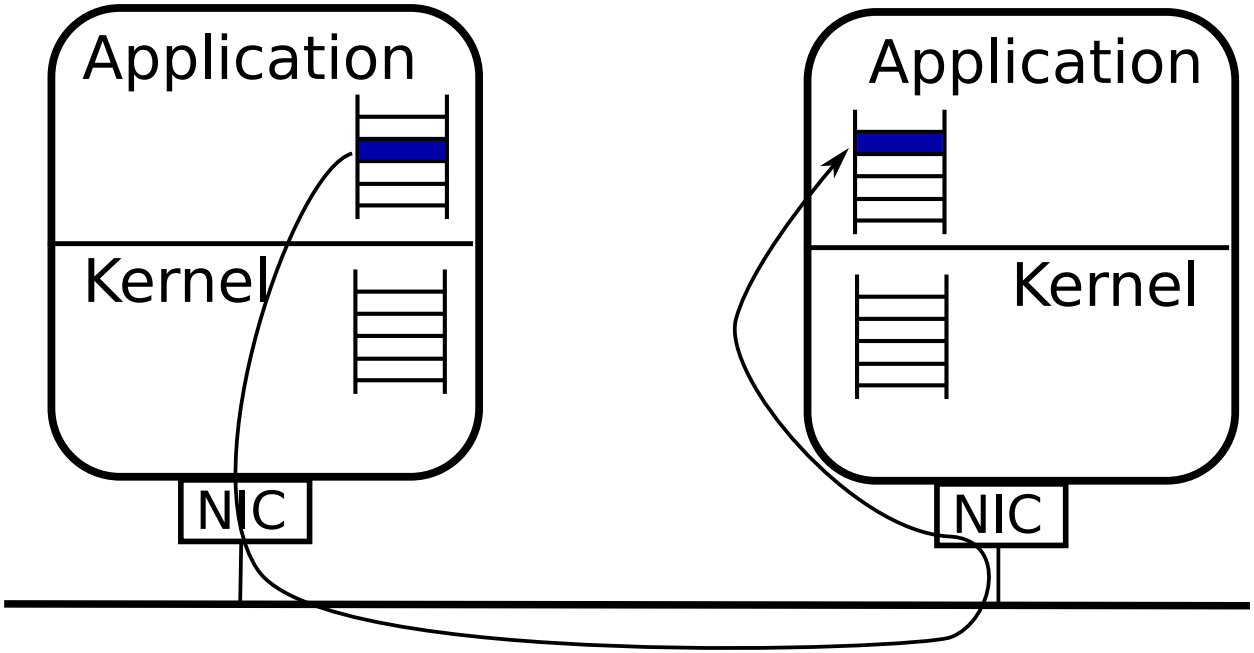
April 13, 2014

In-Memory Distributed Systems

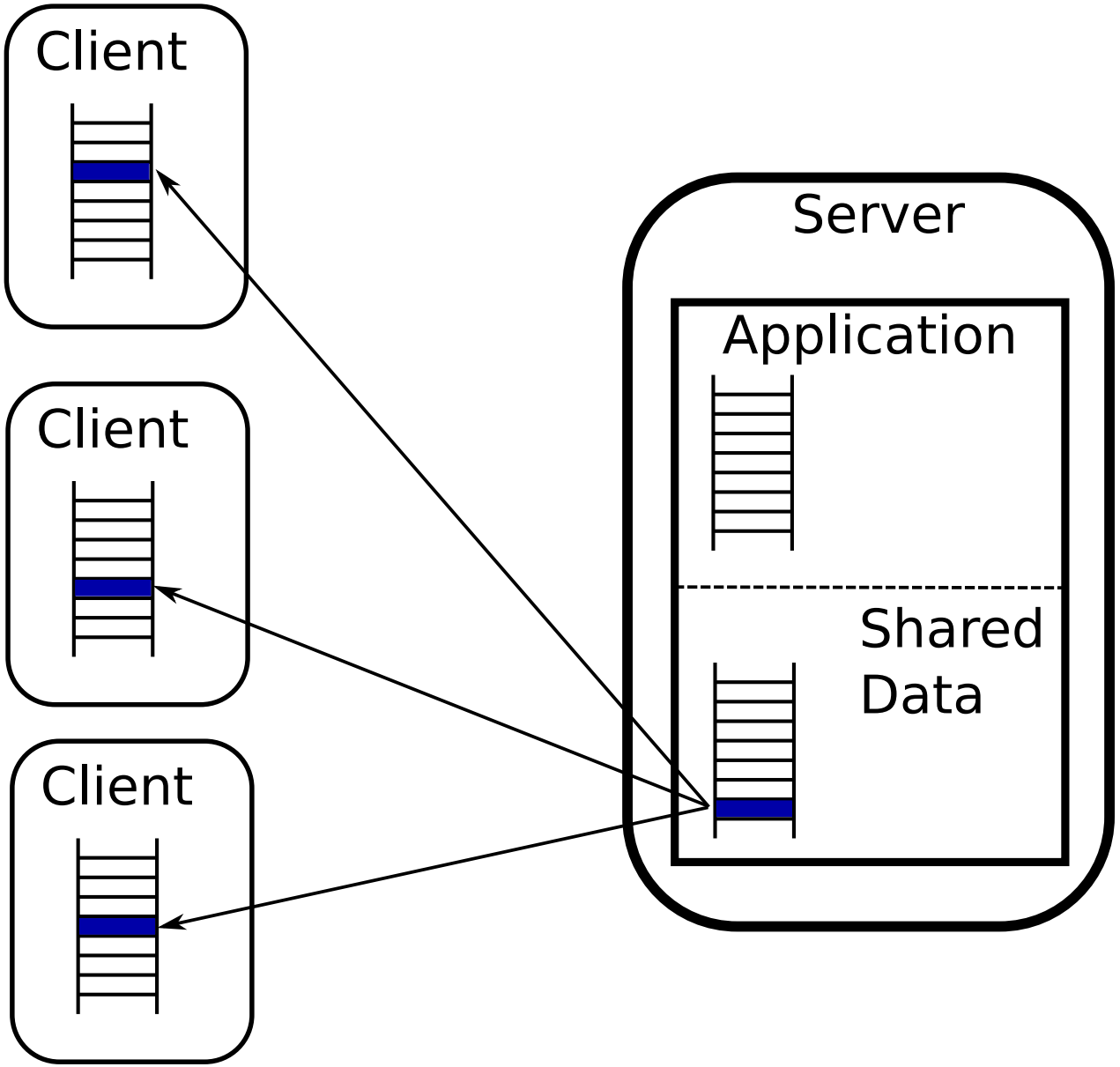
TCP Data Transfer



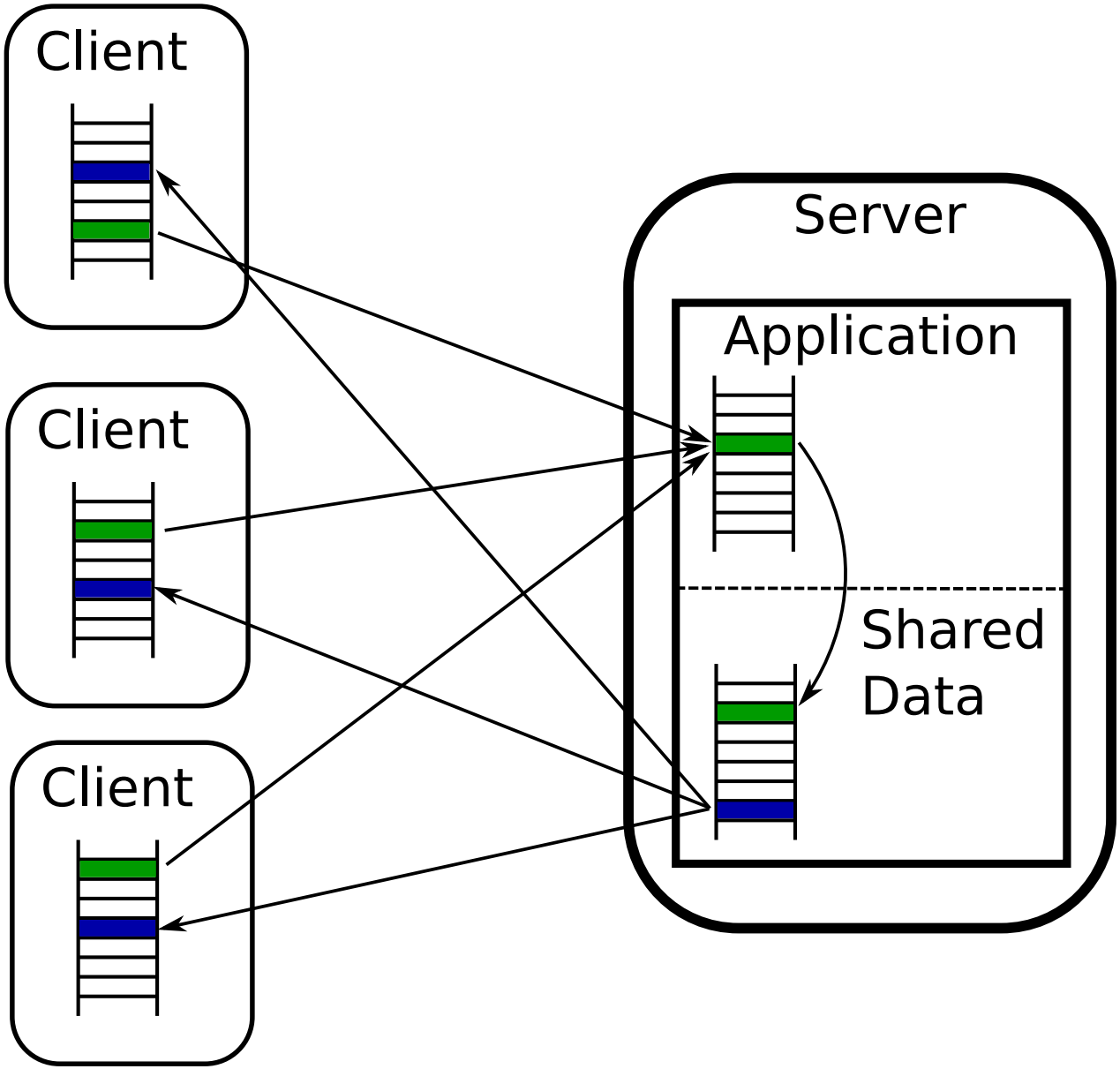
RDMA Data Transfer



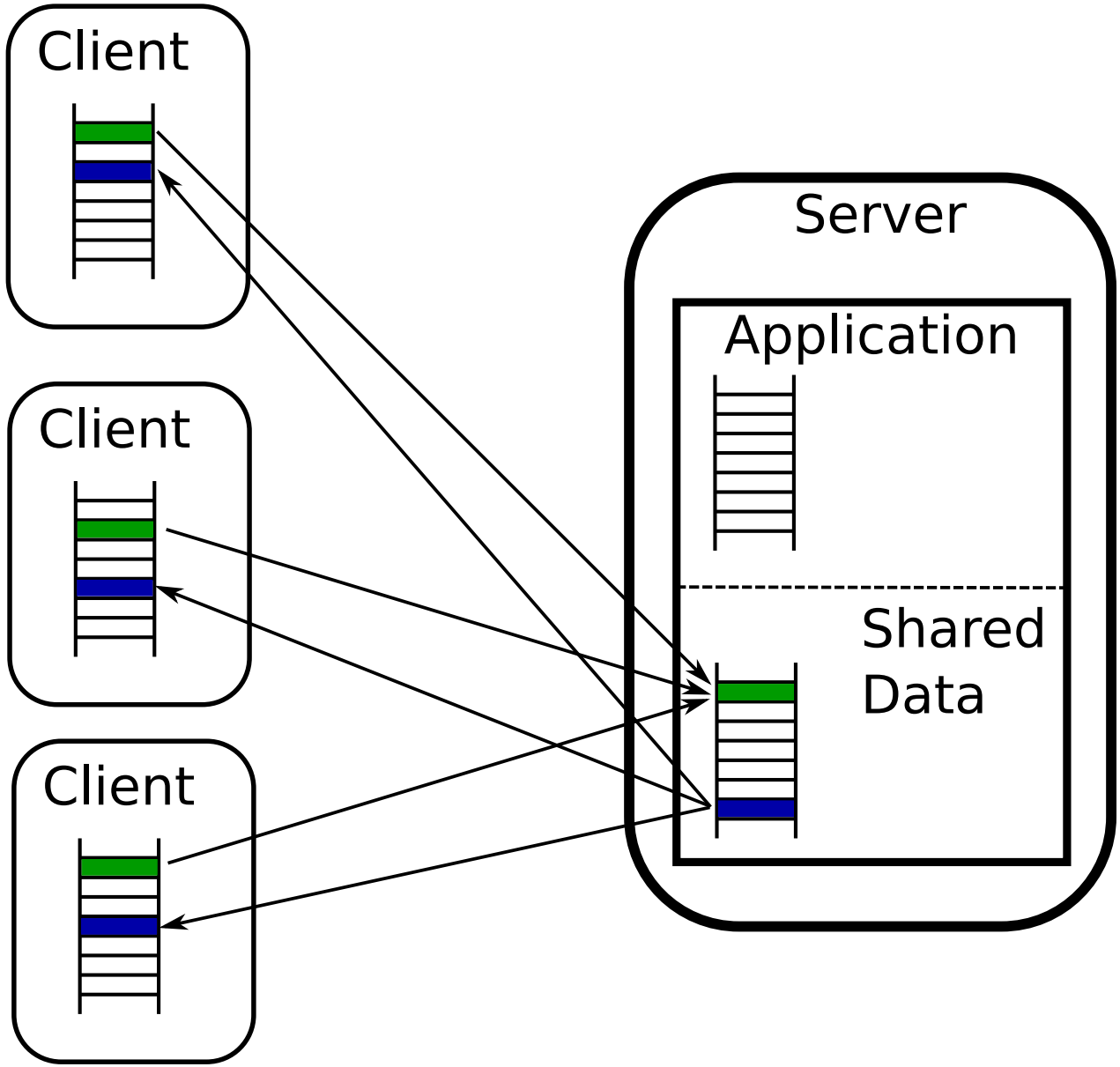
Current State of the Art: Pilaf/FaRM



Current State of the Art: Pilaf/FaRM



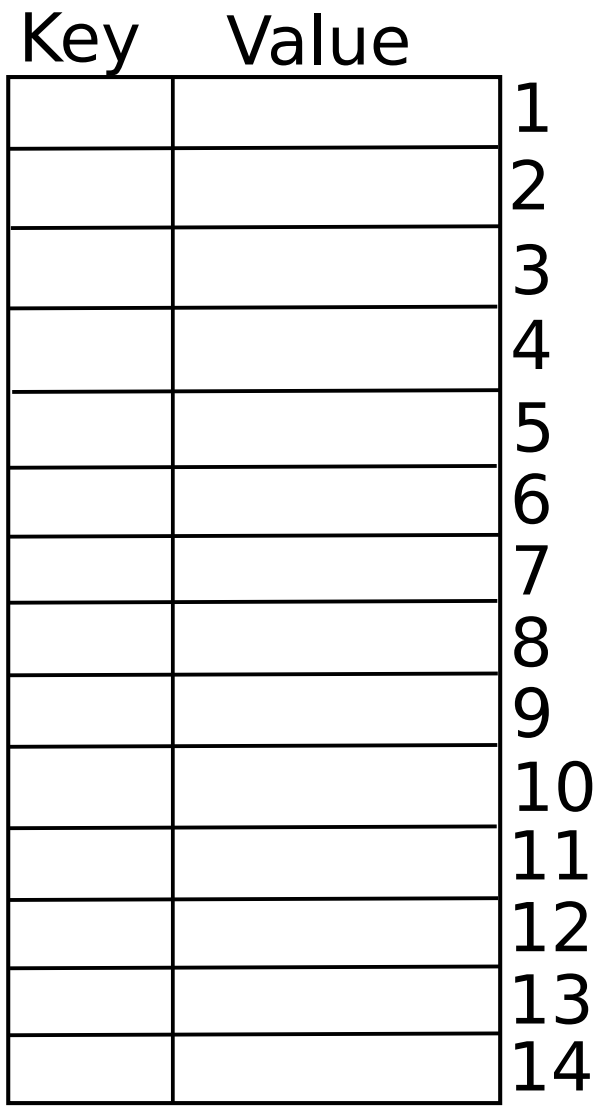
One-Sided RDMA for Reads and Writes



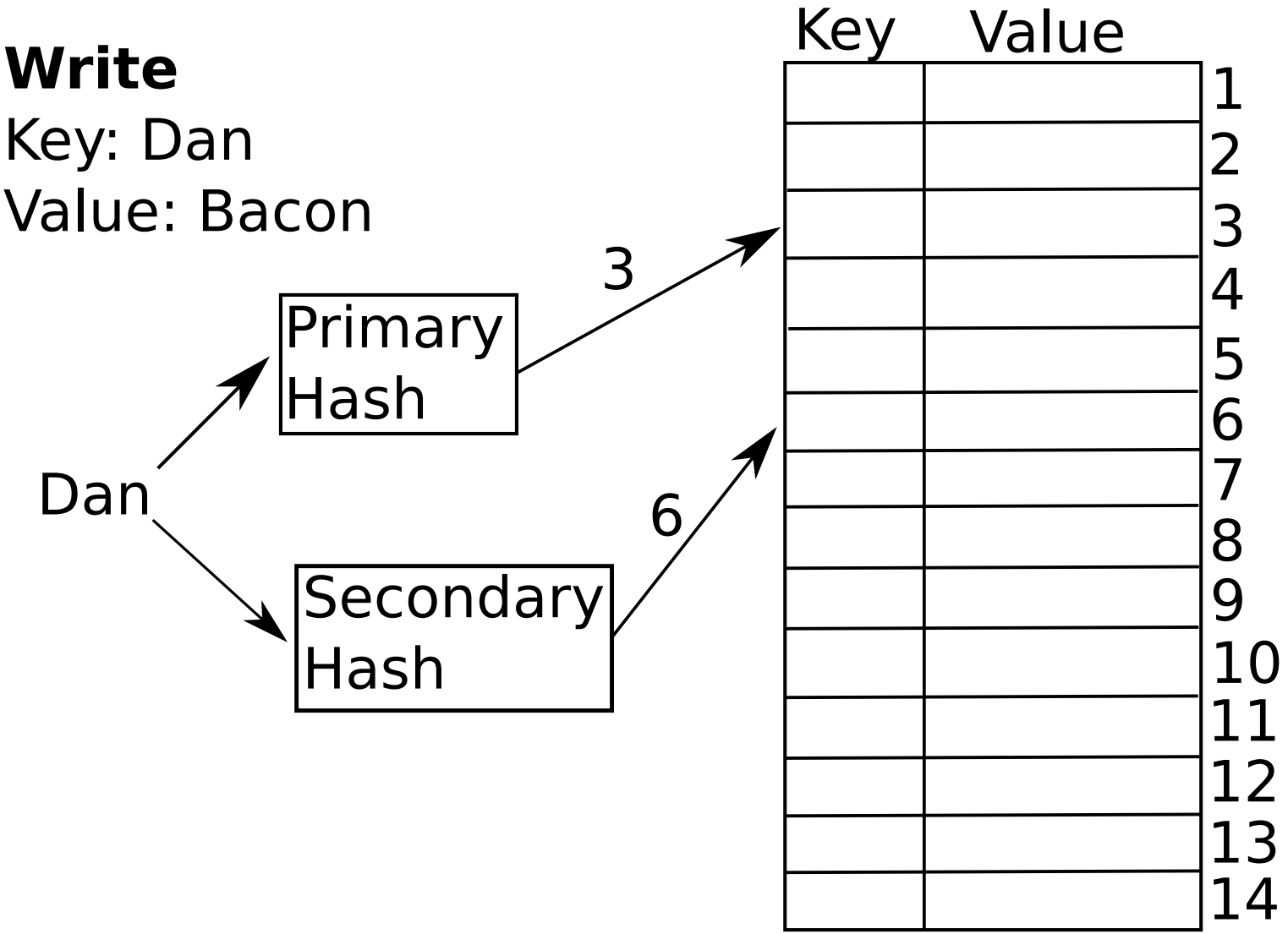
Nessie

- Cuckoo hash table where clients manage both reads and writes
- Design:
  - One-sided RDMA for reads and writes of key-value pairs
  - Checksum to detect corrupted reads
  - RDMA compare-and-swap to create atomic writes
  - Lock-free to eliminate the possibility of deadlock
- Advantage: Full benefits from one-sided RDMA
- Tradeoffs compared to the state of the art:
  - More roundtrips per operation
  - Added complexity

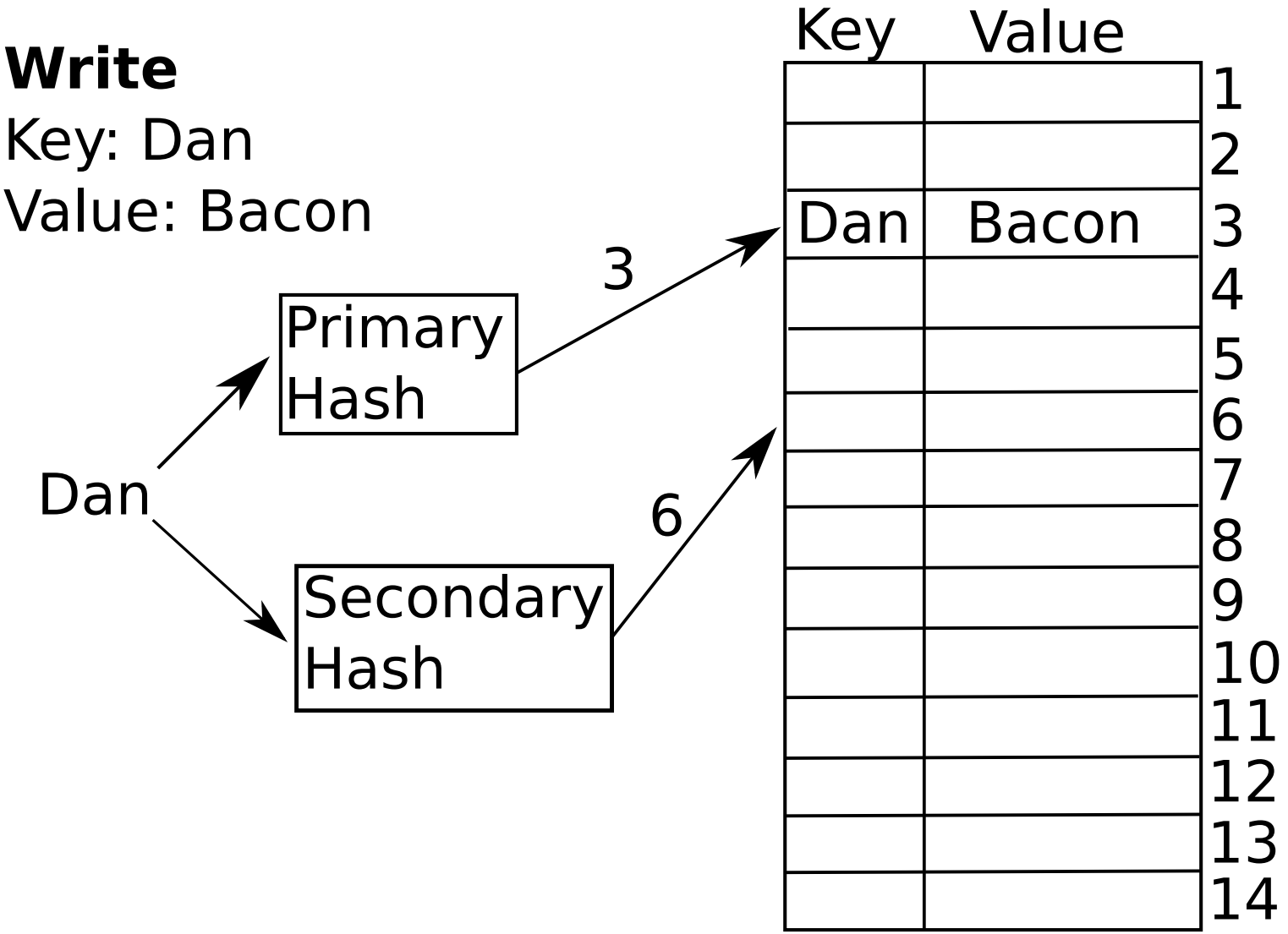
Cuckoo Hash Table



Cuckoo Hash Table

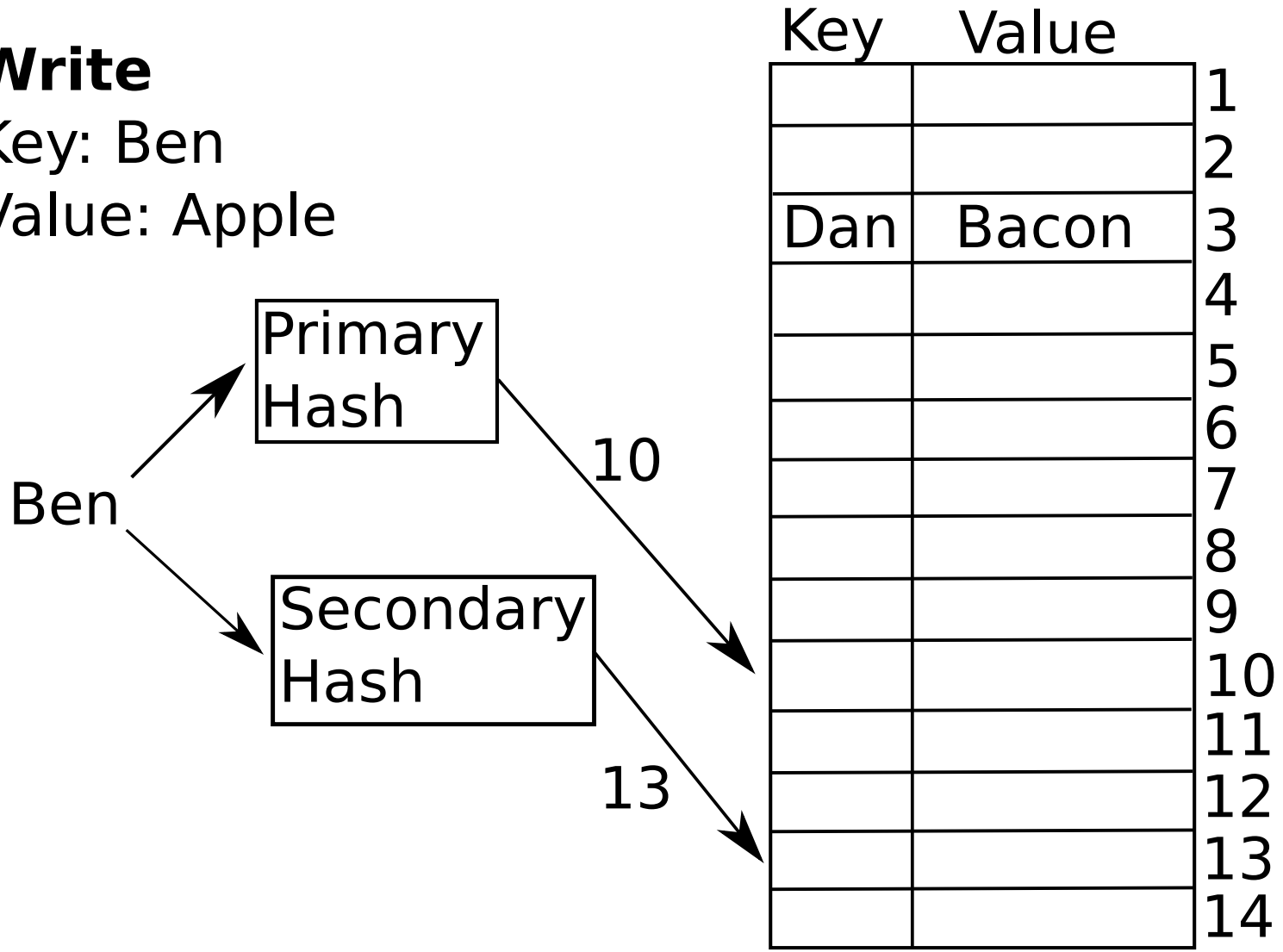


Cuckoo Hash Table



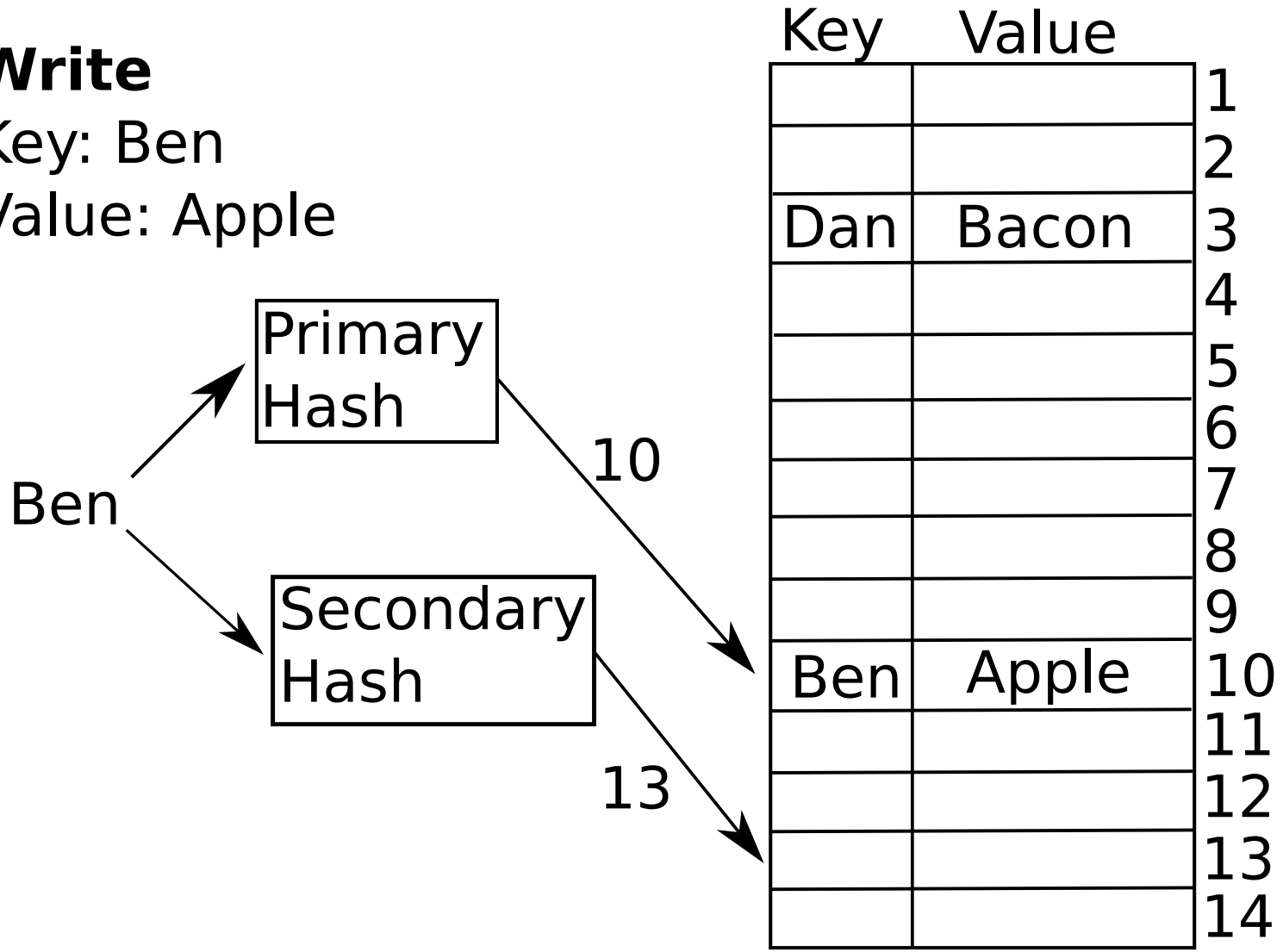
Cuckoo Hash Table

Write
Key: Ben
Value: Apple

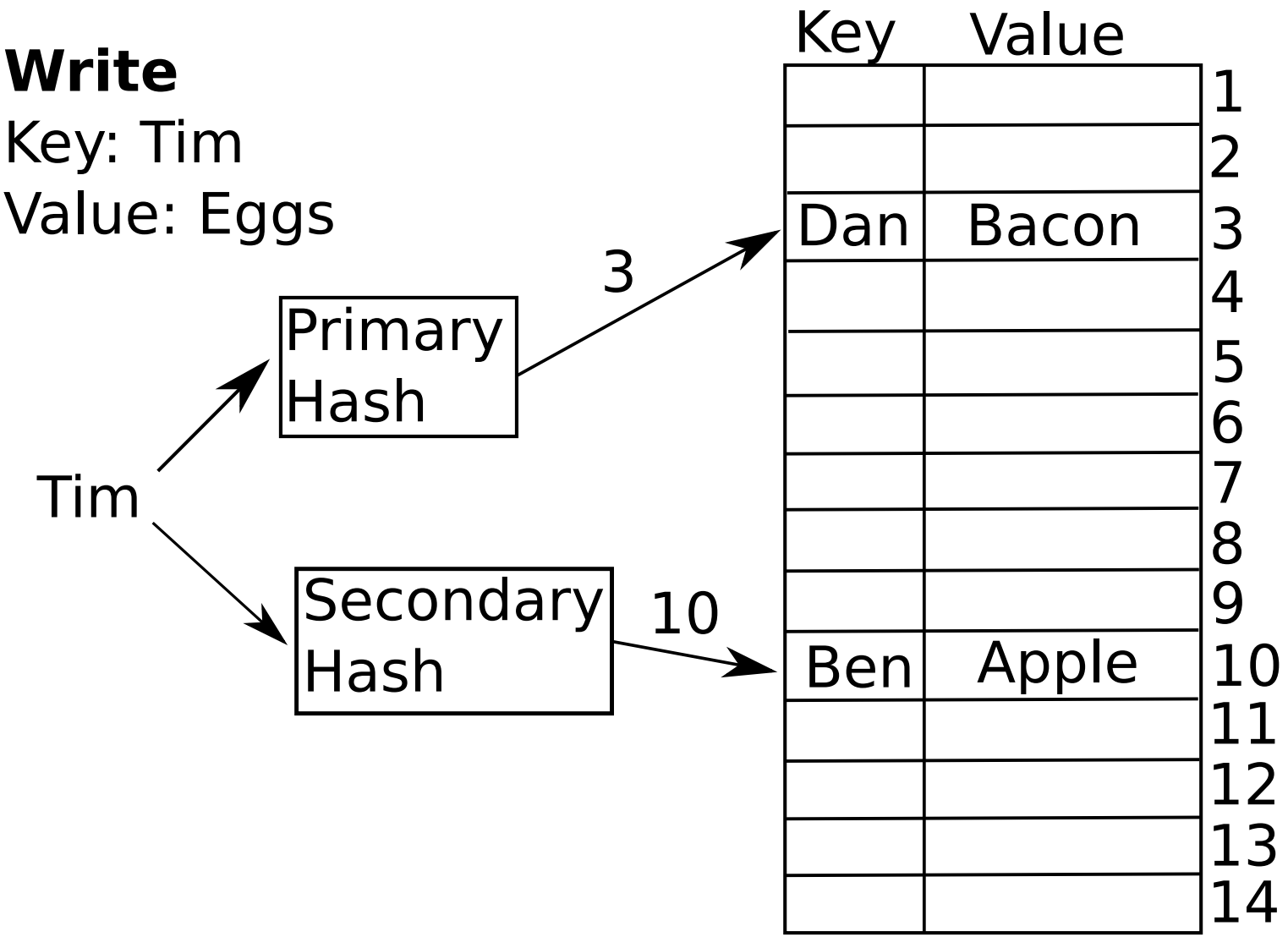


Cuckoo Hash Table

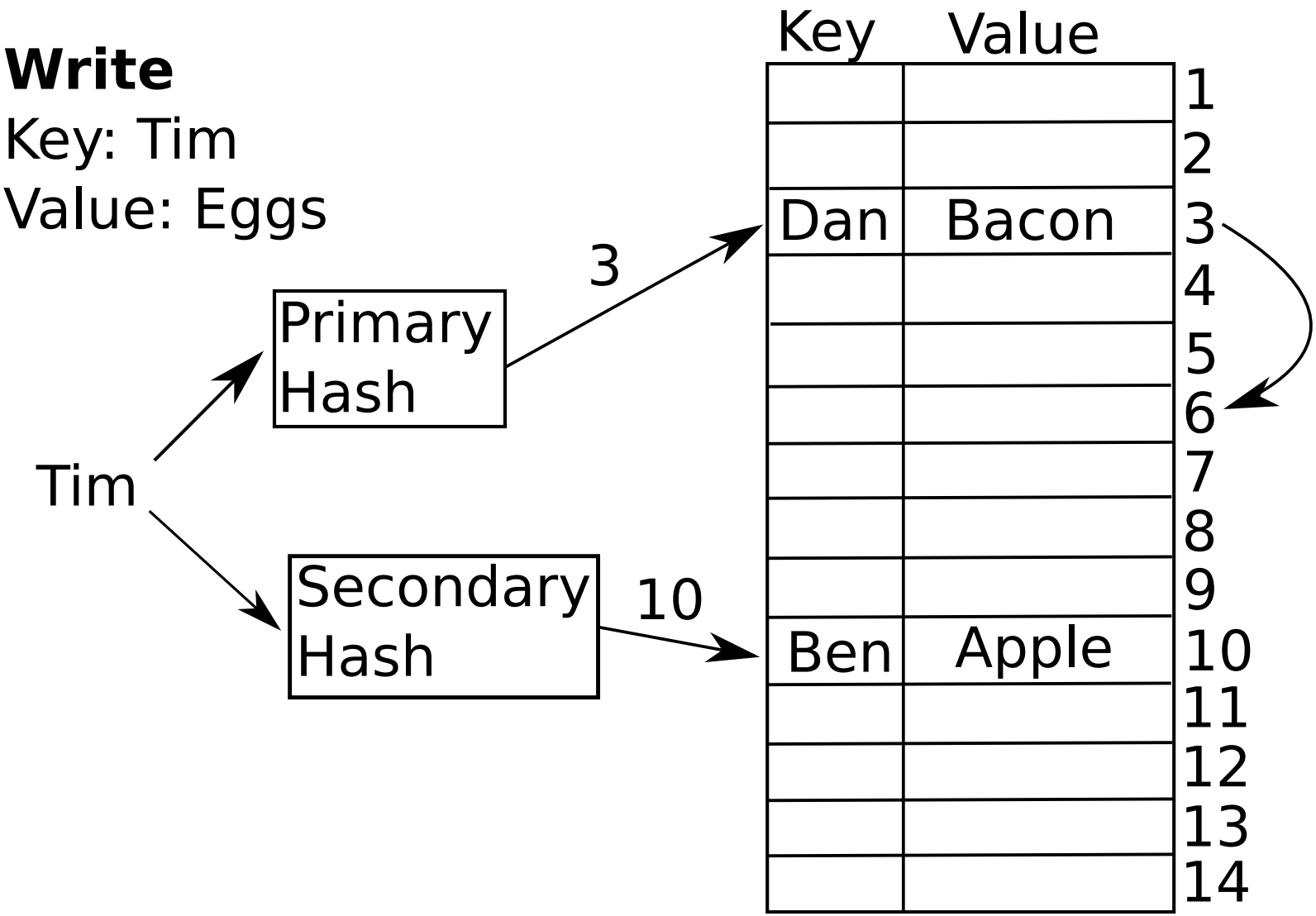
Write
Key: Ben
Value: Apple



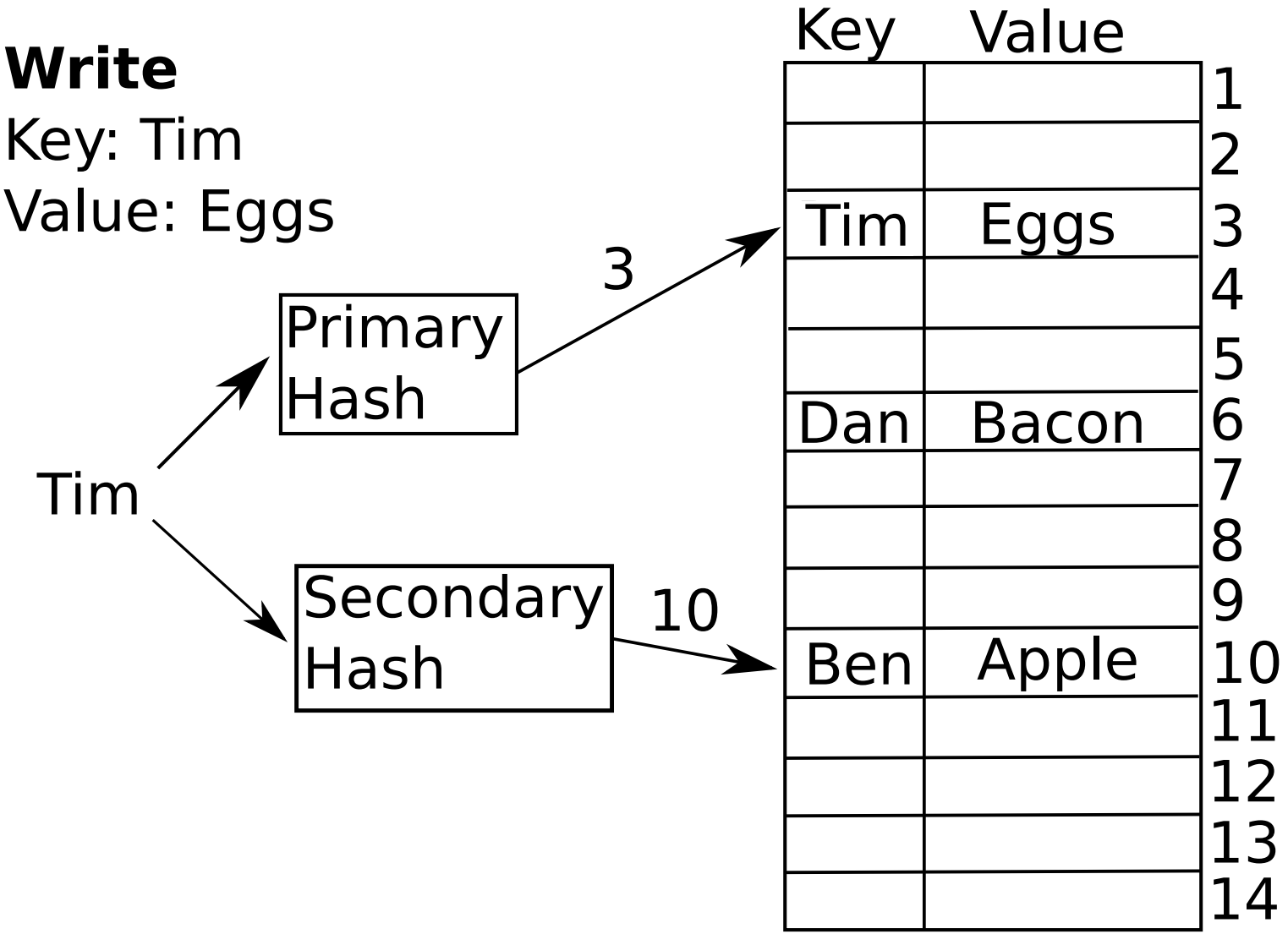
Cuckoo Hash Table



Cuckoo Hash Table

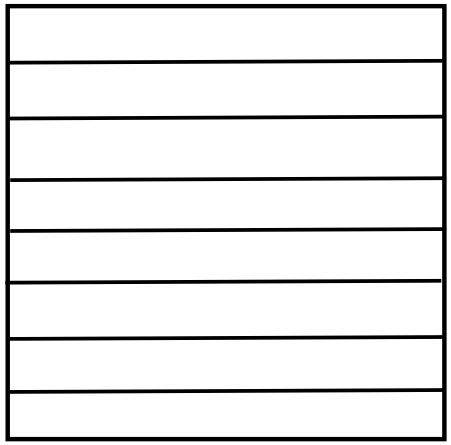


Cuckoo Hash Table

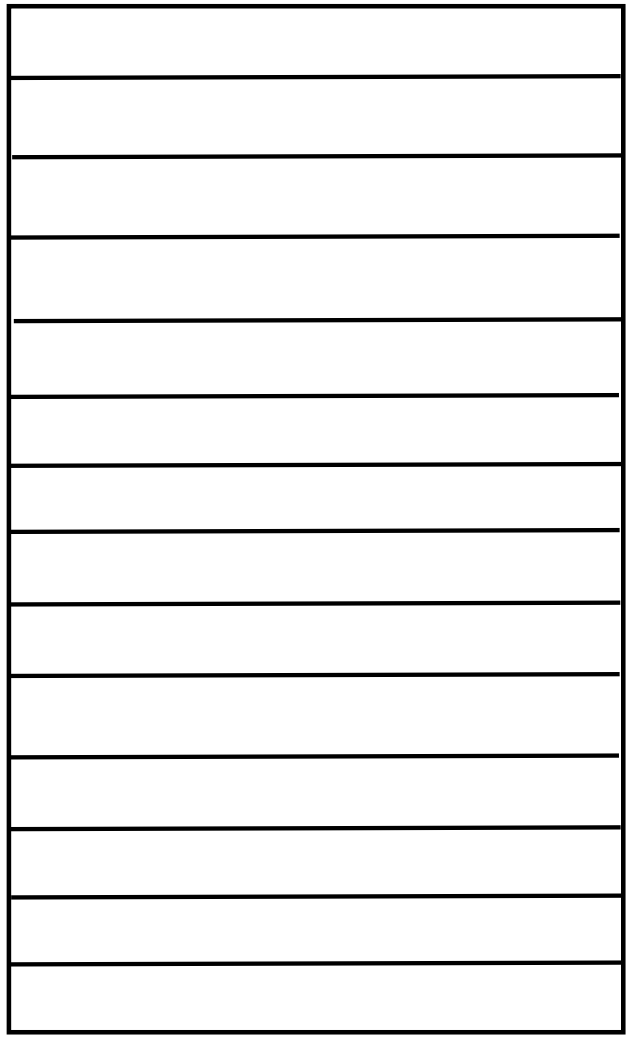


Nessie

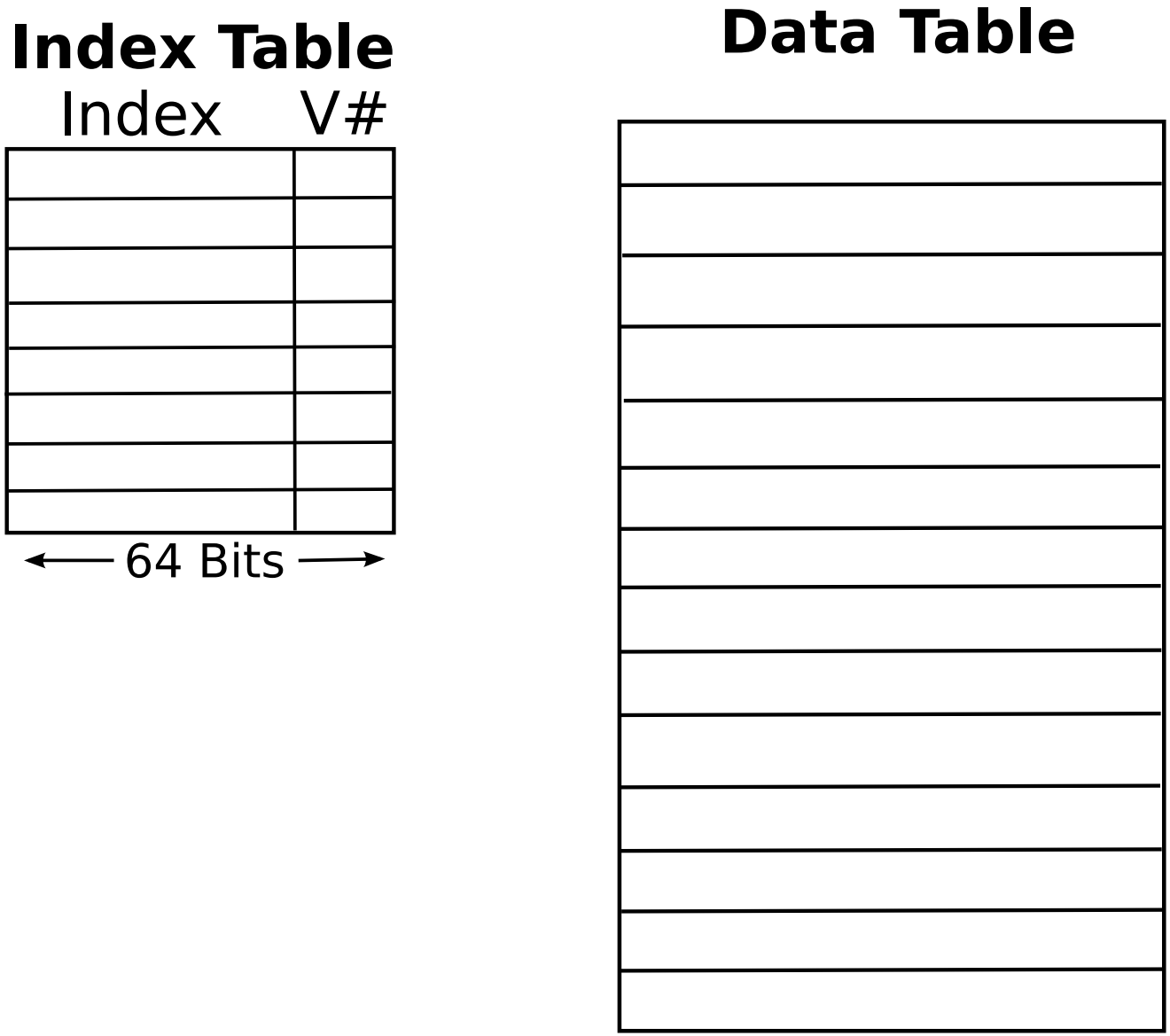
Index Table



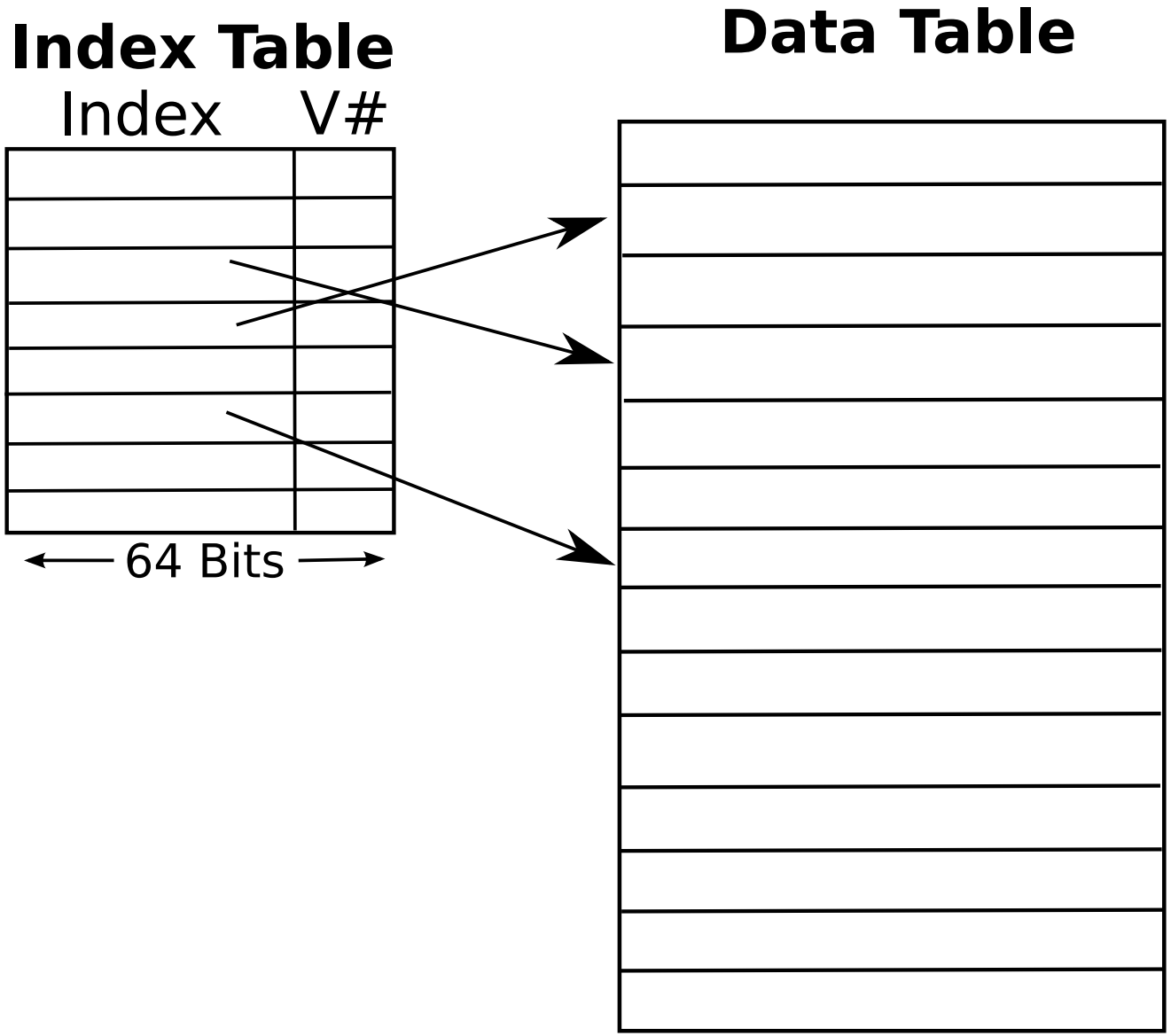
Data Table



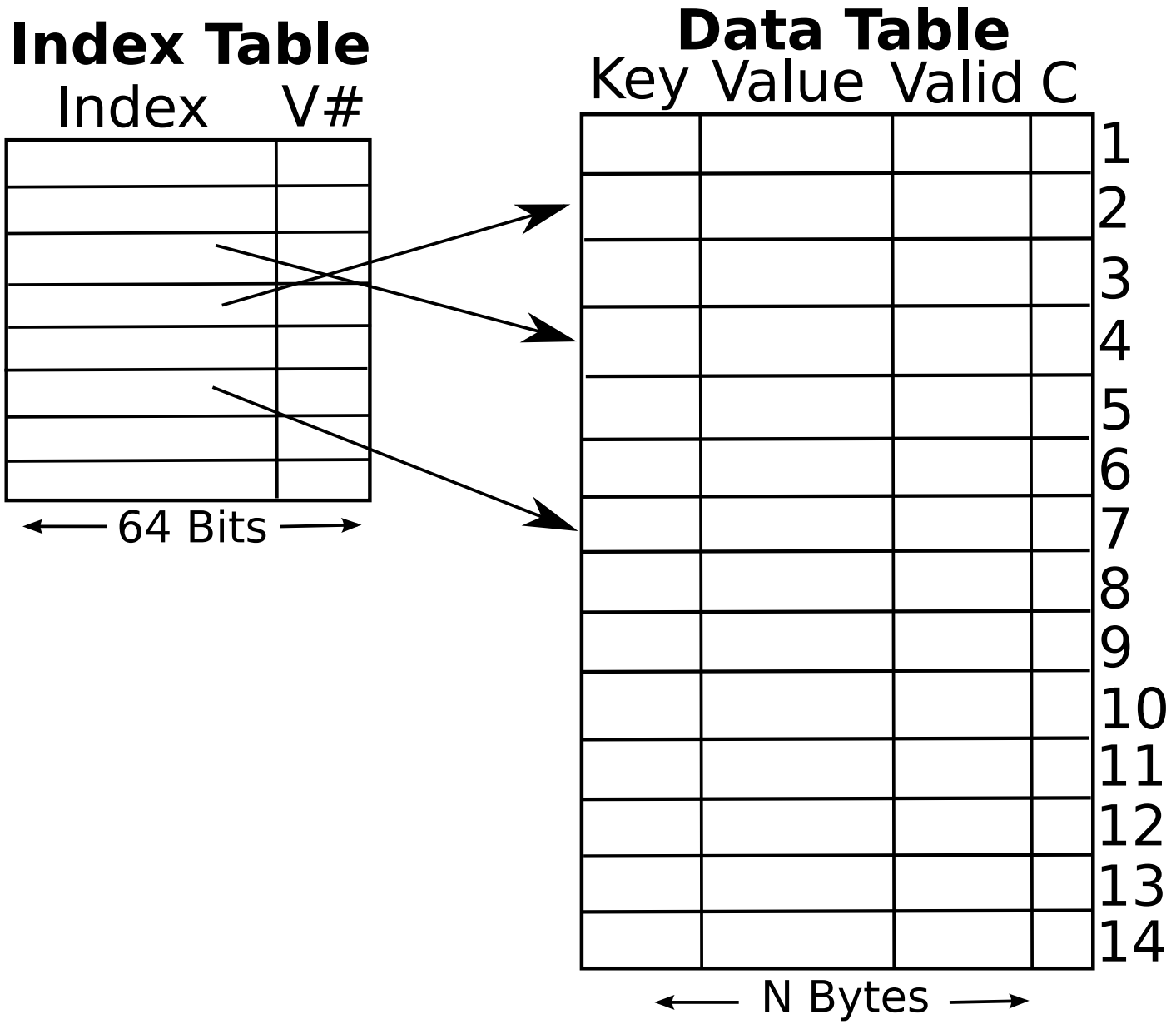
Nessie



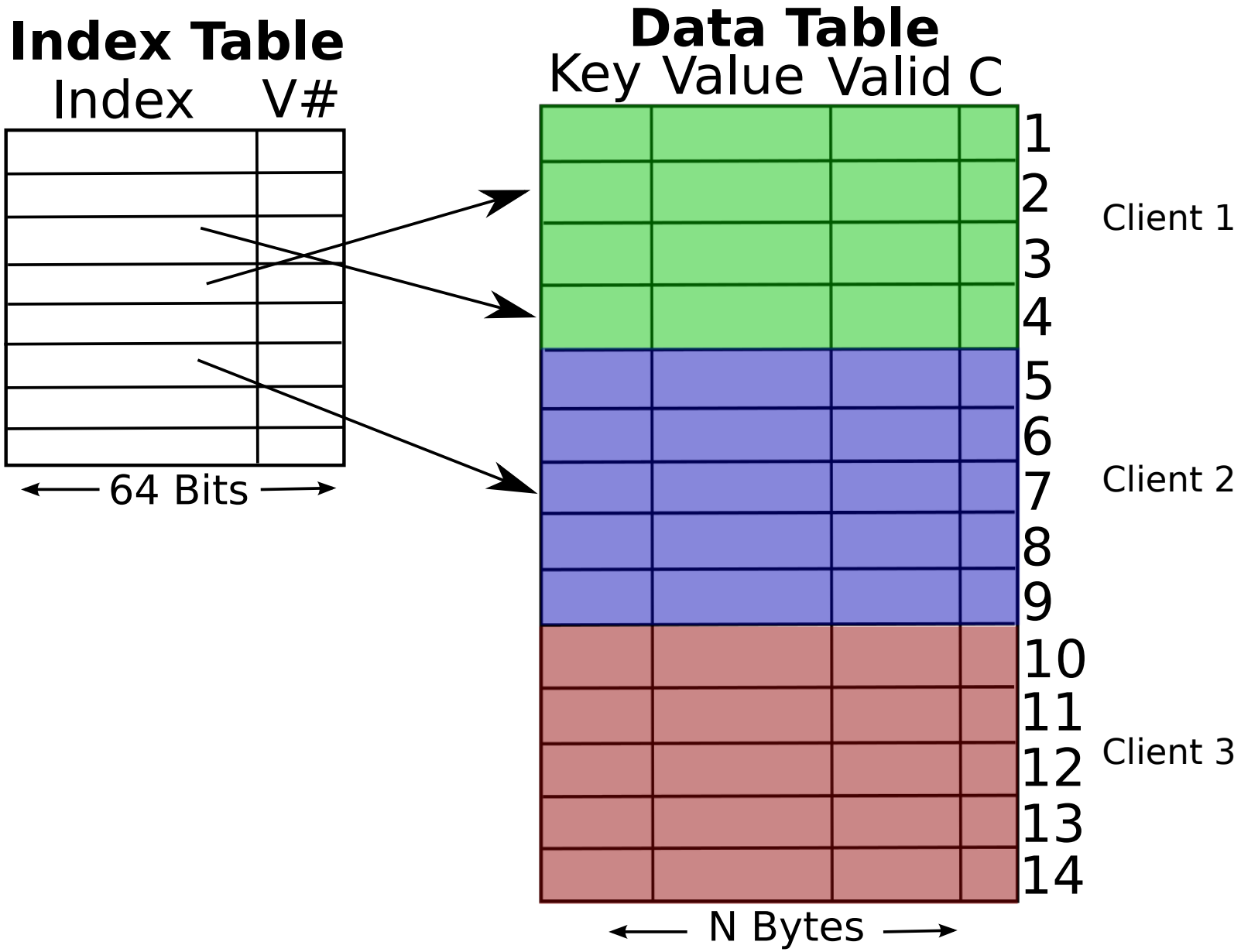
Nessie



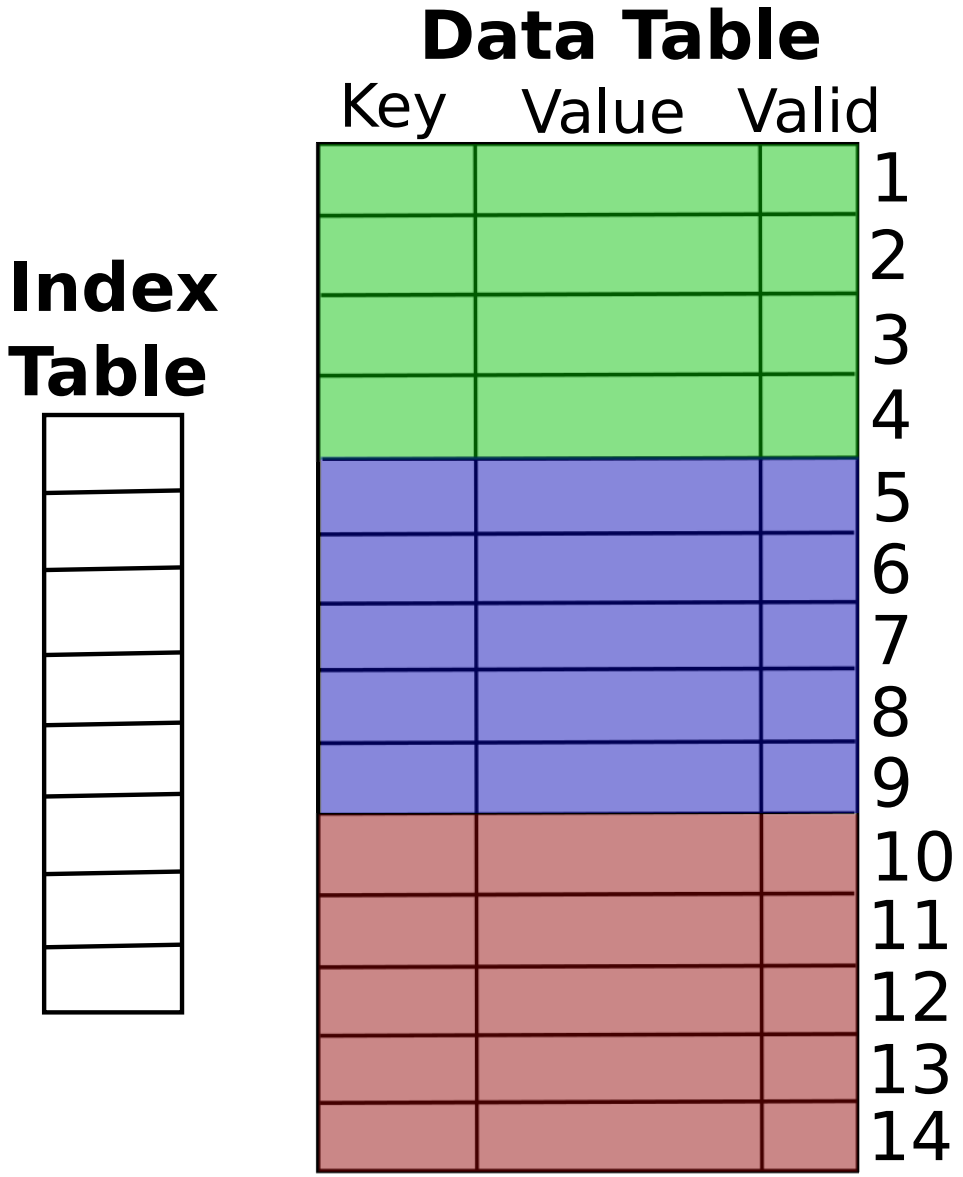
Nessie


Index Table
Index
V#
64 Bits
Data Table
Key
Value
Valid
C
1
2
3
4
5
6
7
8
9
10
11
12
13
14
N Bytes

Nessie


Index Table
Index V#
64 Bits
Data Table
Key Value Valid C
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Client 1
Client 2
Client 3
N Bytes

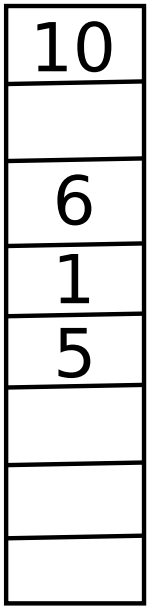
Nessie


Data Table
Key
Value
Valid
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Index Table

Nessie: Write

Write
Key: Tim
Value: Eggs

Index Table



Data Table

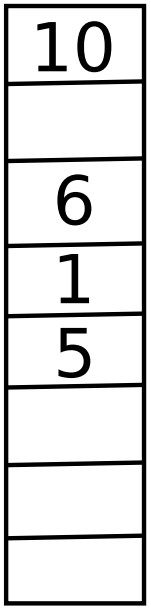
	Key	Value	Valid
1	Sam	Pizza	T
2			
3			
4			
5	Tim	Carrot	T
6	Dan	Bacon	T
7			
8			
9			
10	Ben	Apple	T
11			
12			
13			
14			

Nessie: Write

Write Data

Write
Key: Tim
Value: Eggs

Index Table



Data Table

	Key	Value	Valid
1	Sam	Pizza	T
2			
3			
4			
5	Tim	Carrot	T
6	Dan	Bacon	T
7	**Tim**	**Eggs**	**F**
8			
9			
10	Ben	Apple	T
11			
12			
13			
14			

Nessie: Write

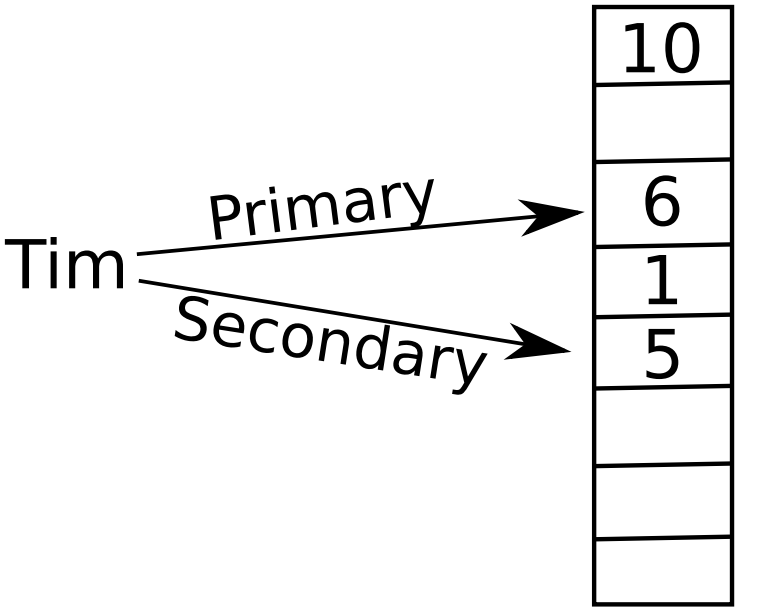
Read Index Table

Write

Key: Tim

Value: Eggs

Index Table



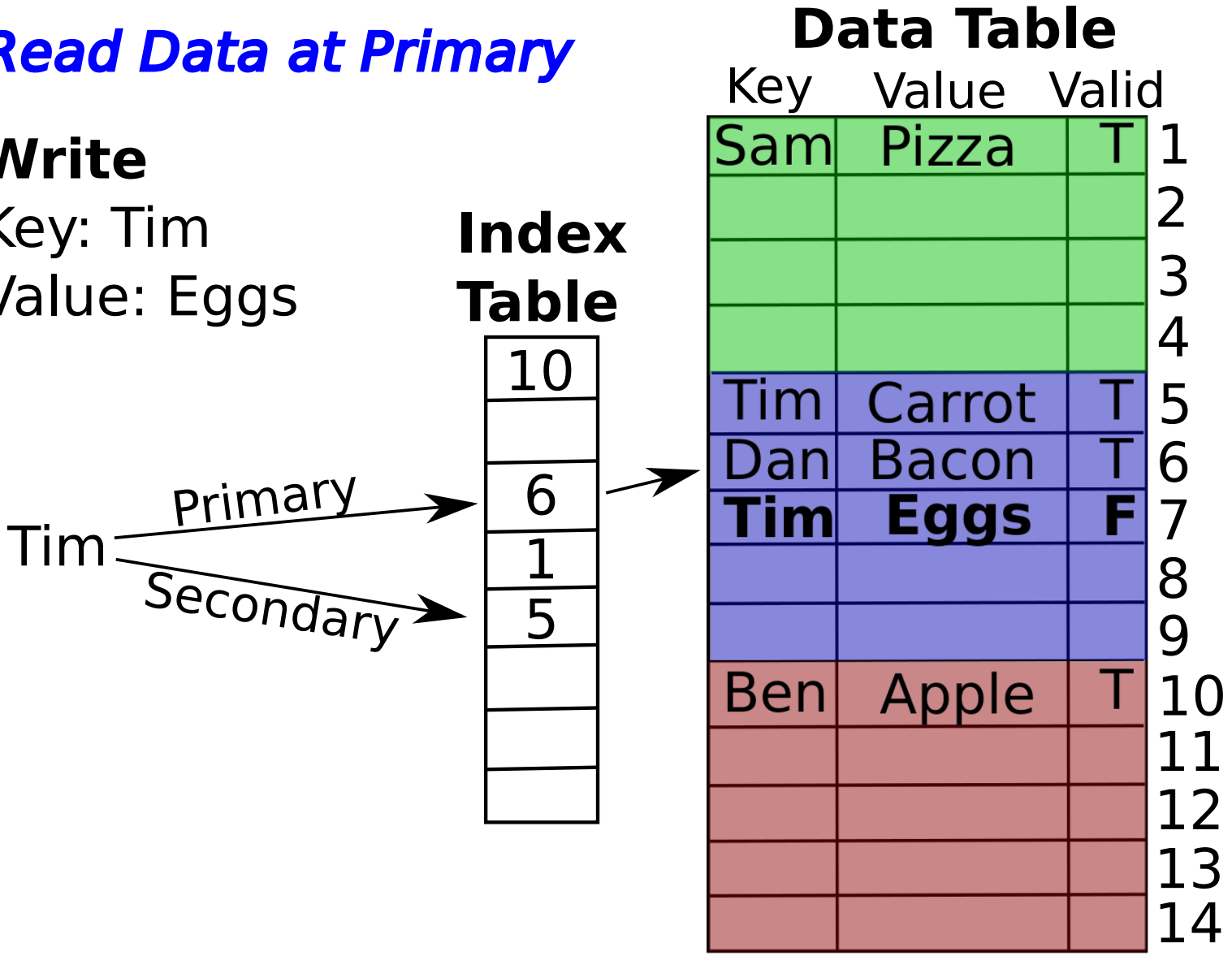
Data Table

	Key	Value	Valid
1	Sam	Pizza	T
2			
3			
4			
5	Tim	Carrot	T
6	Dan	Bacon	T
7	**Tim**	**Eggs**	**F**
8			
9			
10	Ben	Apple	T
11			
12			
13			
14			

Nessie: Write

Read Data at Primary

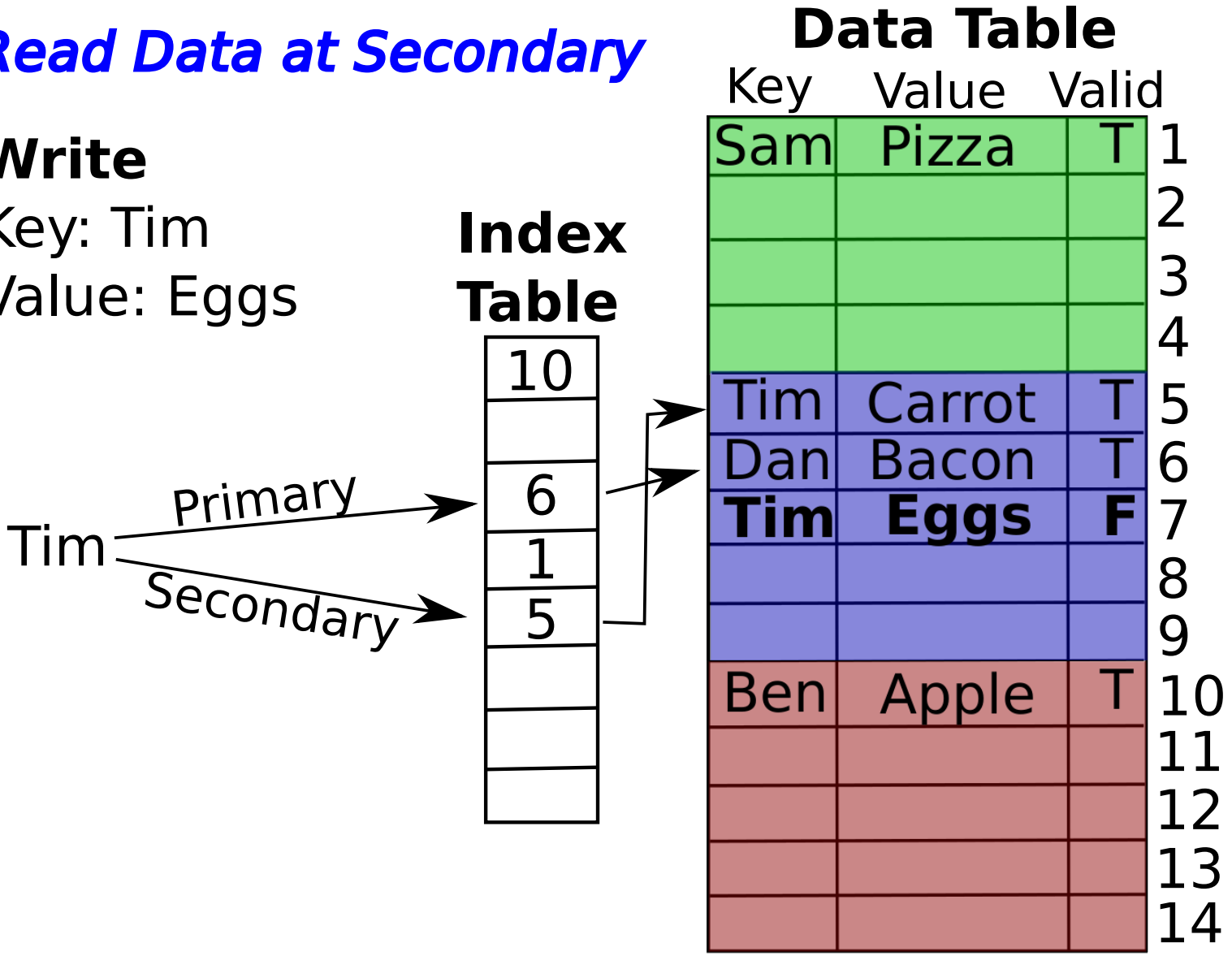
Write
Key: Tim
Value: Eggs



Nessie: Write

Read Data at Secondary

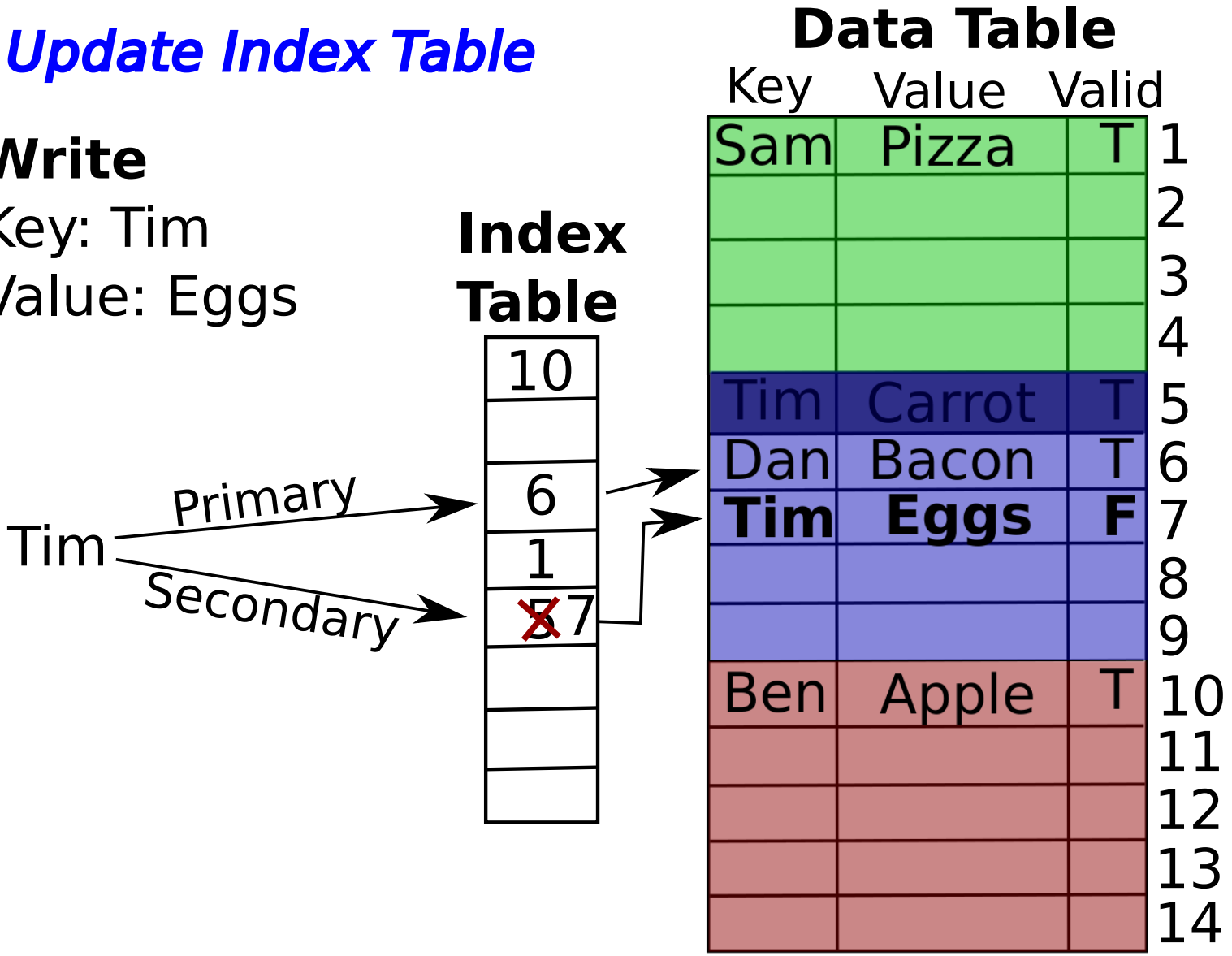
Write
Key: Tim
Value: Eggs



Nessie: Write

Update Index Table

Write
Key: Tim
Value: Eggs



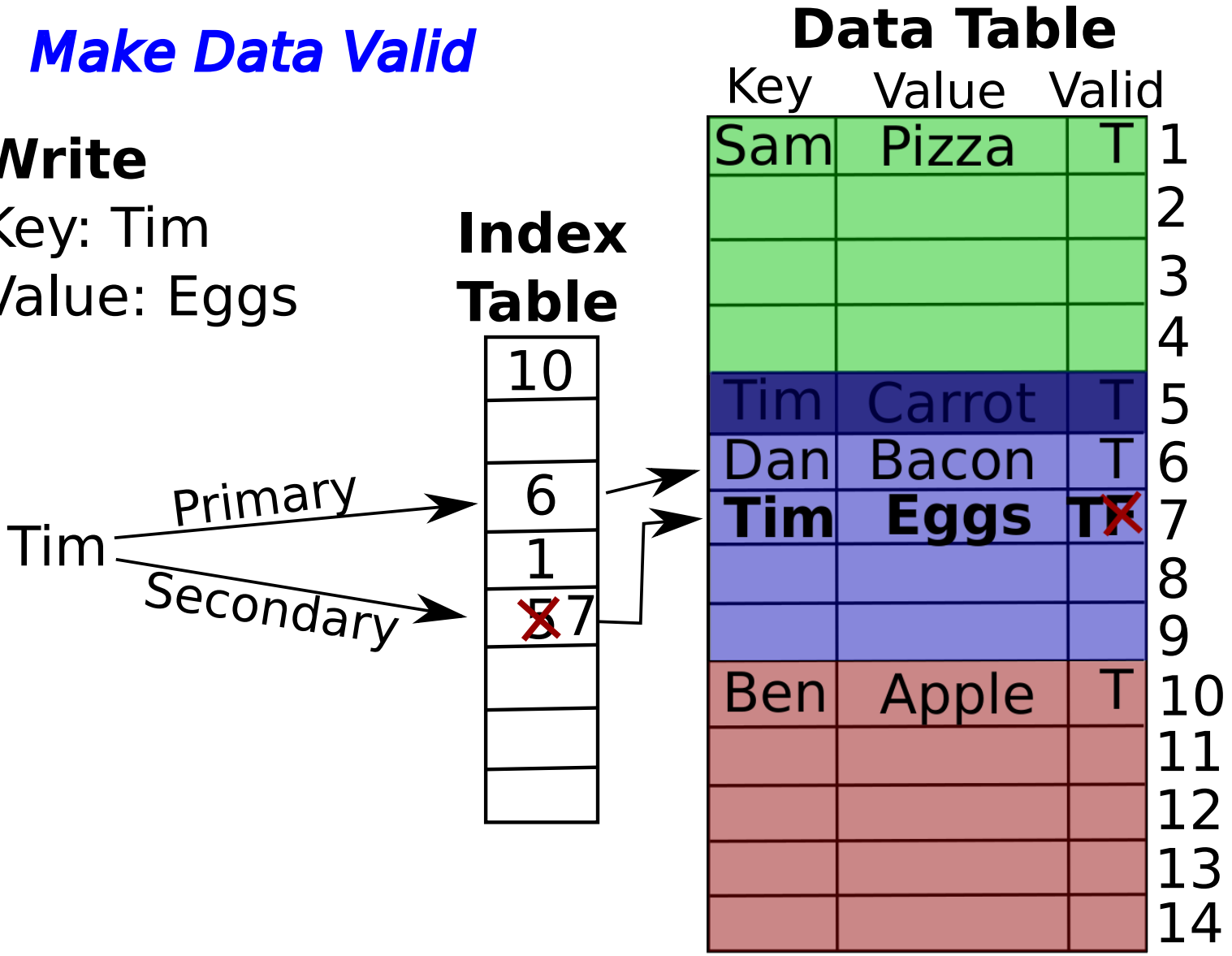
Nessie: Write

Make Data Valid

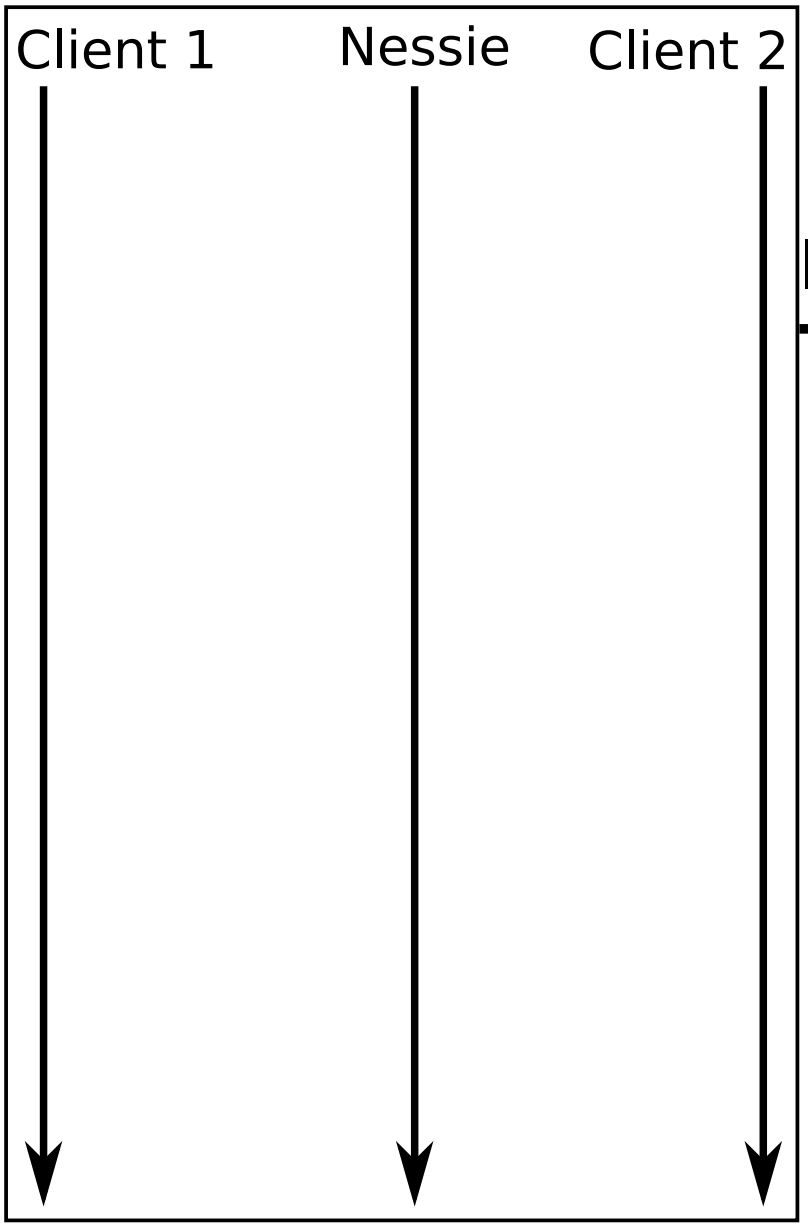
Write

Key: Tim

Value: Eggs



Nessie: Write with Conflict



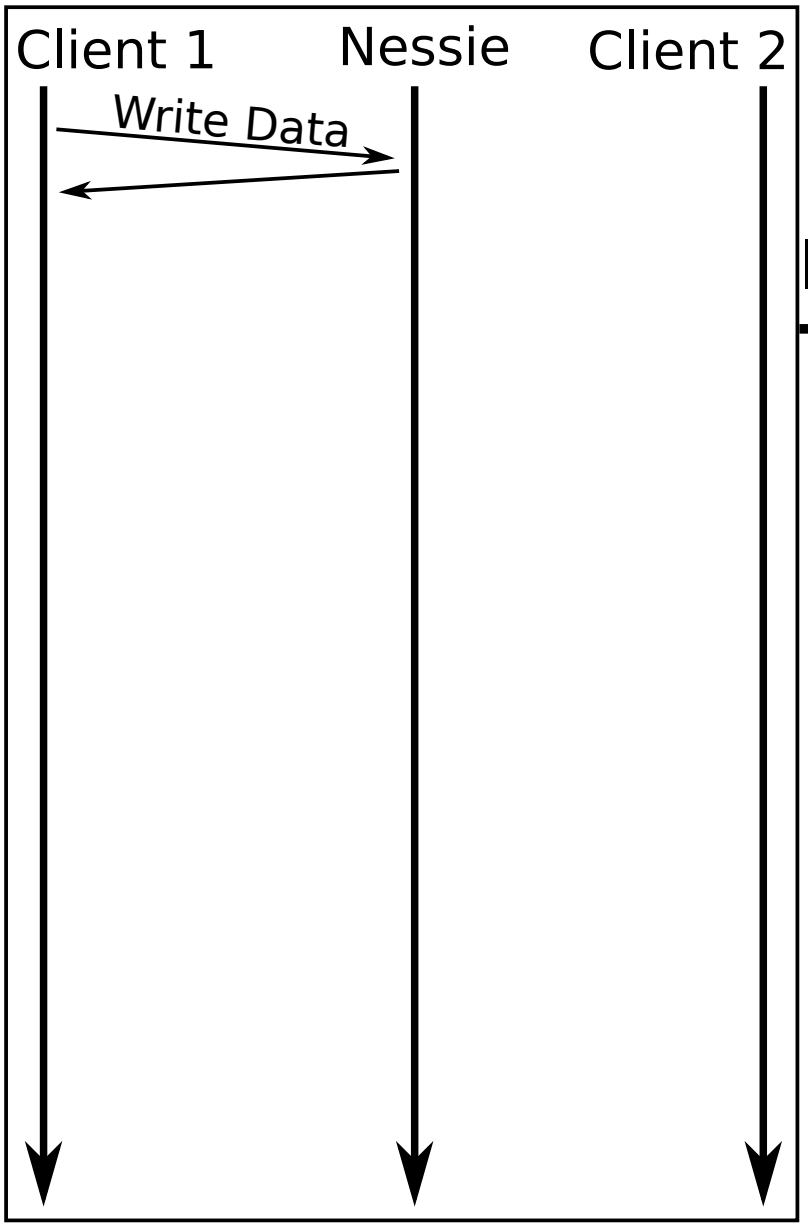
Index Table

10
6
1
5

Data Table

	Key	Value	Valid
1	Sam	Pizza	T
2			
3			
4			
5	Tim	Carrot	T
6	Dan	Bacon	T
7			
8			
9			
10	Ben	Apple	T
11			
12			
13			
14			

Nessie: Write with Conflict



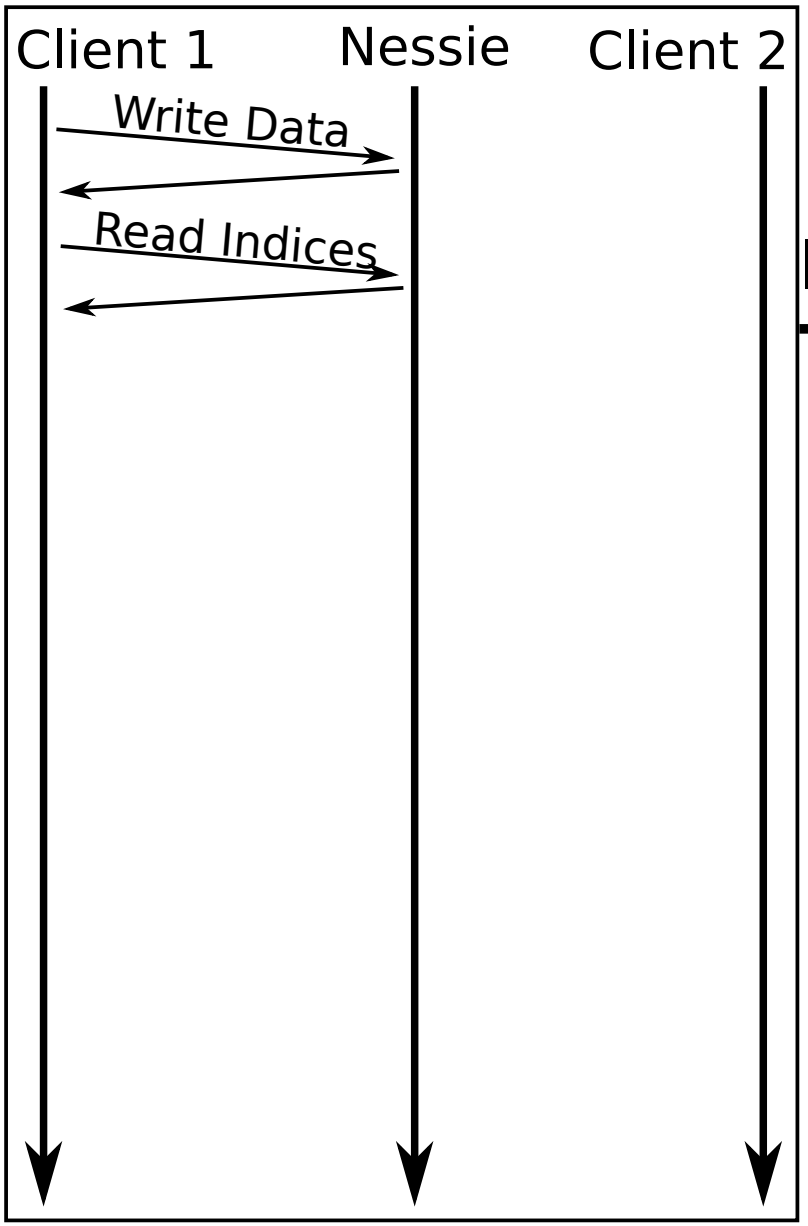
Index Table

10
6
1
5

Data Table

	Key	Value	Valid
1	Sam	Pizza	T
2			
3			
4			
5	Tim	Carrot	T
6	Dan	Bacon	T
7	**Tim**	**Eggs**	**F**
8			
9			
10	Ben	Apple	T
11			
12			
13			
14			

Nessie: Write with Conflict



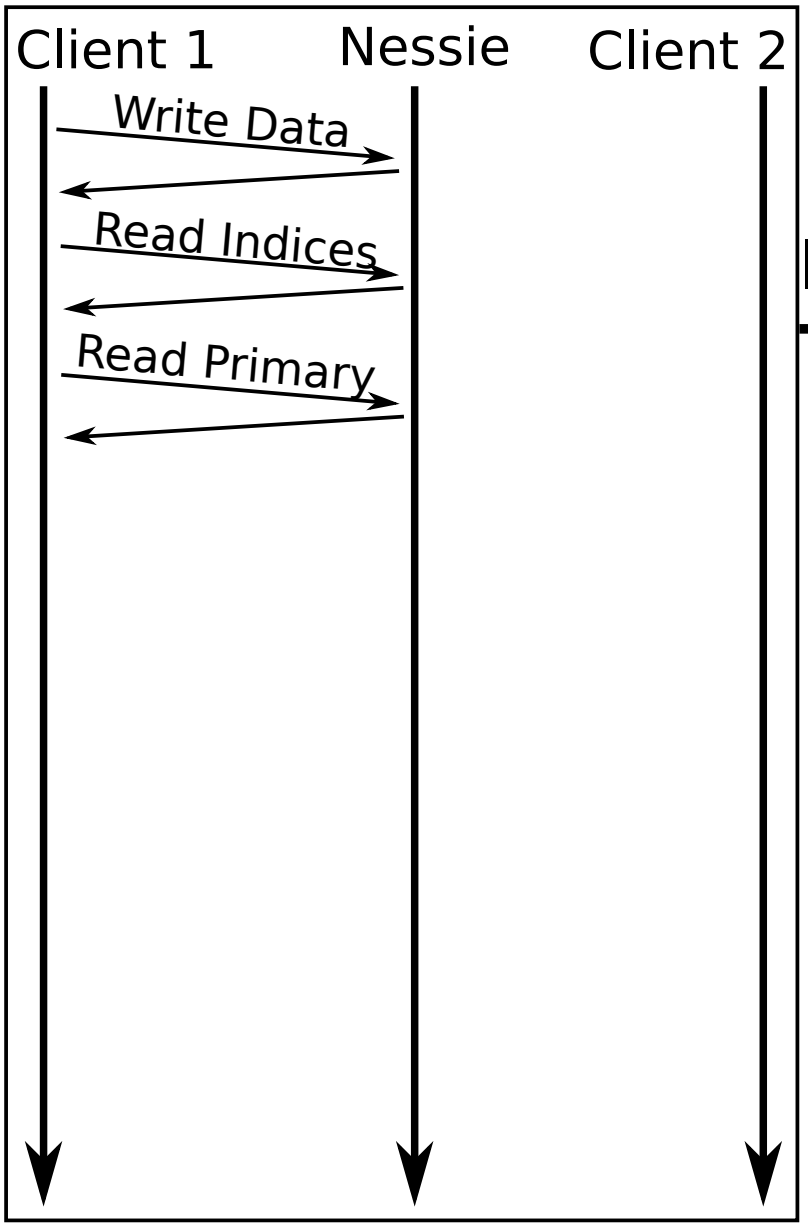
Index Table

10
6
1
5

Data Table

Key	Value	Valid	
Sam	Pizza	T	1
			2
			3
			4
Tim	Carrot	T	5
Dan	Bacon	T	6
Tim	**Eggs**	**F**	7
			8
			9
Ben	Apple	T	10
			11
			12
			13
			14

Nessie: Write with Conflict



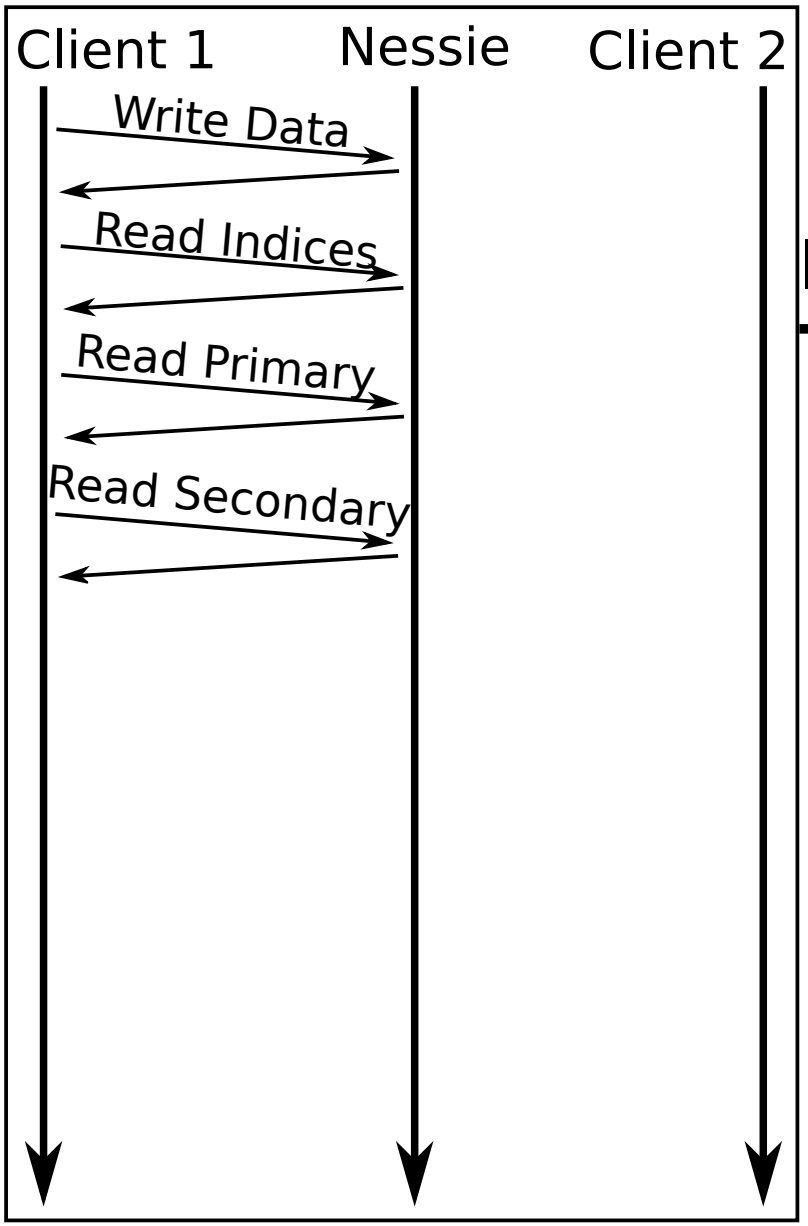
Index Table

10
6
1
5

Data Table

Key	Value	Valid	
Sam	Pizza	T	1
			2
			3
			4
Tim	Carrot	T	5
Dan	Bacon	T	6
Tim	**Eggs**	**F**	7
			8
			9
Ben	Apple	T	10
			11
			12
			13
			14

Nessie: Write with Conflict



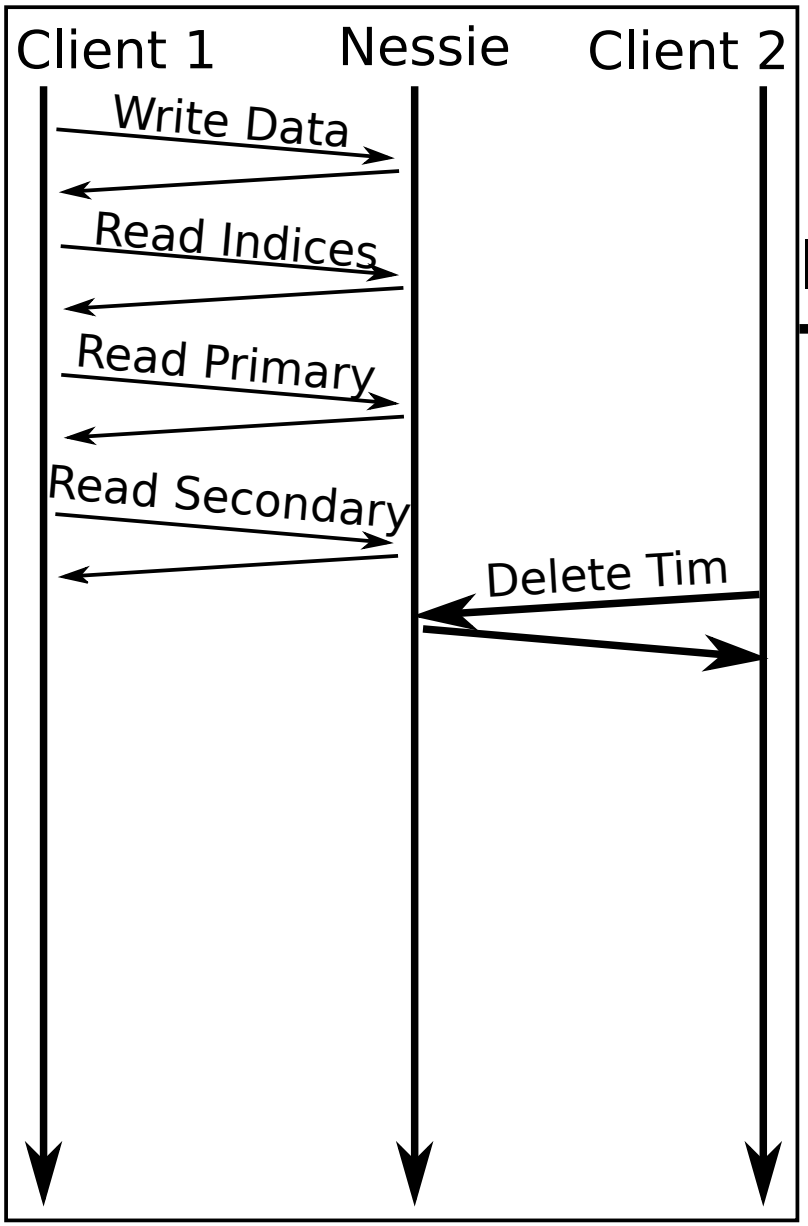
Index Table

10
6
1
5

Data Table

	Key	Value	Valid
1	Sam	Pizza	T
2			
3			
4			
5	Tim	Carrot	T
6	Dan	Bacon	T
7	**Tim**	**Eggs**	**F**
8			
9			
10	Ben	Apple	T
11			
12			
13			
14			

Nessie: Write with Conflict



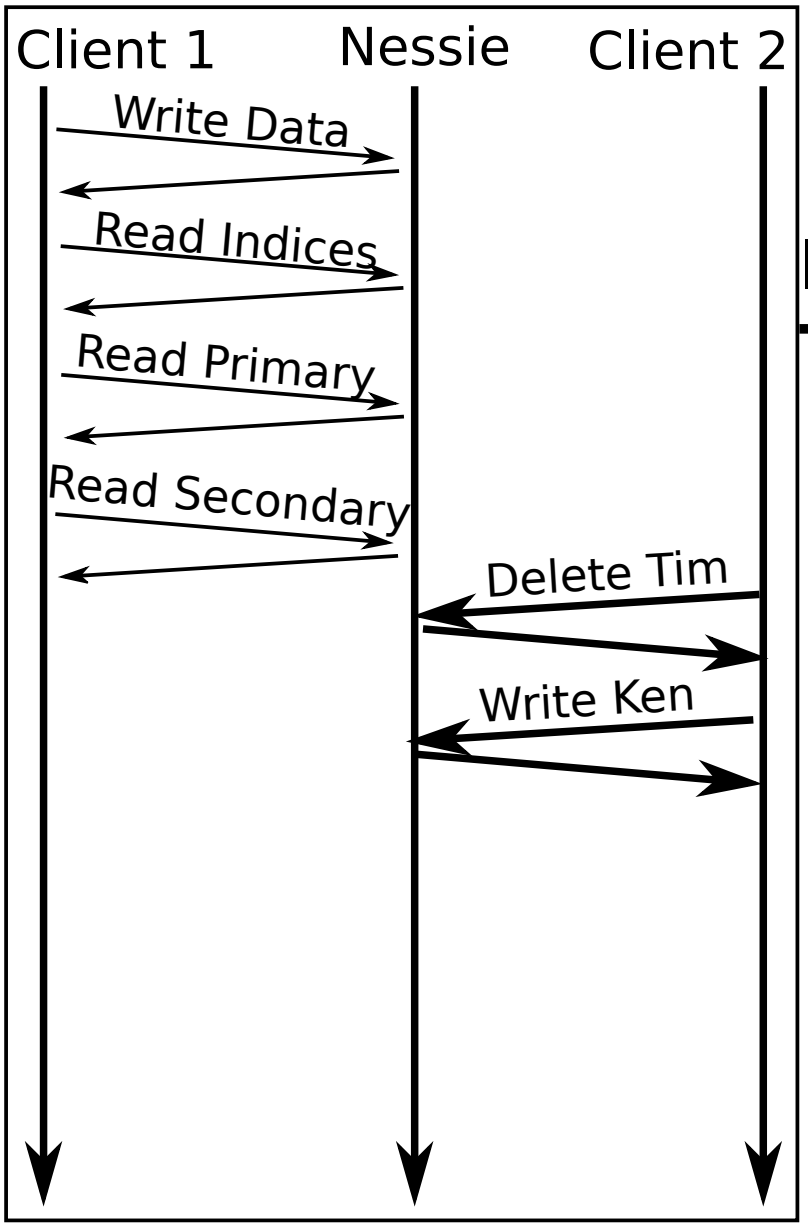
Index Table

10
6
1
~~5~~ 0

Data Table

Key	Value	Valid	
Sam	Pizza	T	1
			2
			3
			4
Tim	Carrot	T	5
Dan	Bacon	T	6
Tim	**Eggs**	**F**	7
			8
			9
Ben	Apple	T	10
			11
			12
			13
			14

Nessie: Write with Conflict



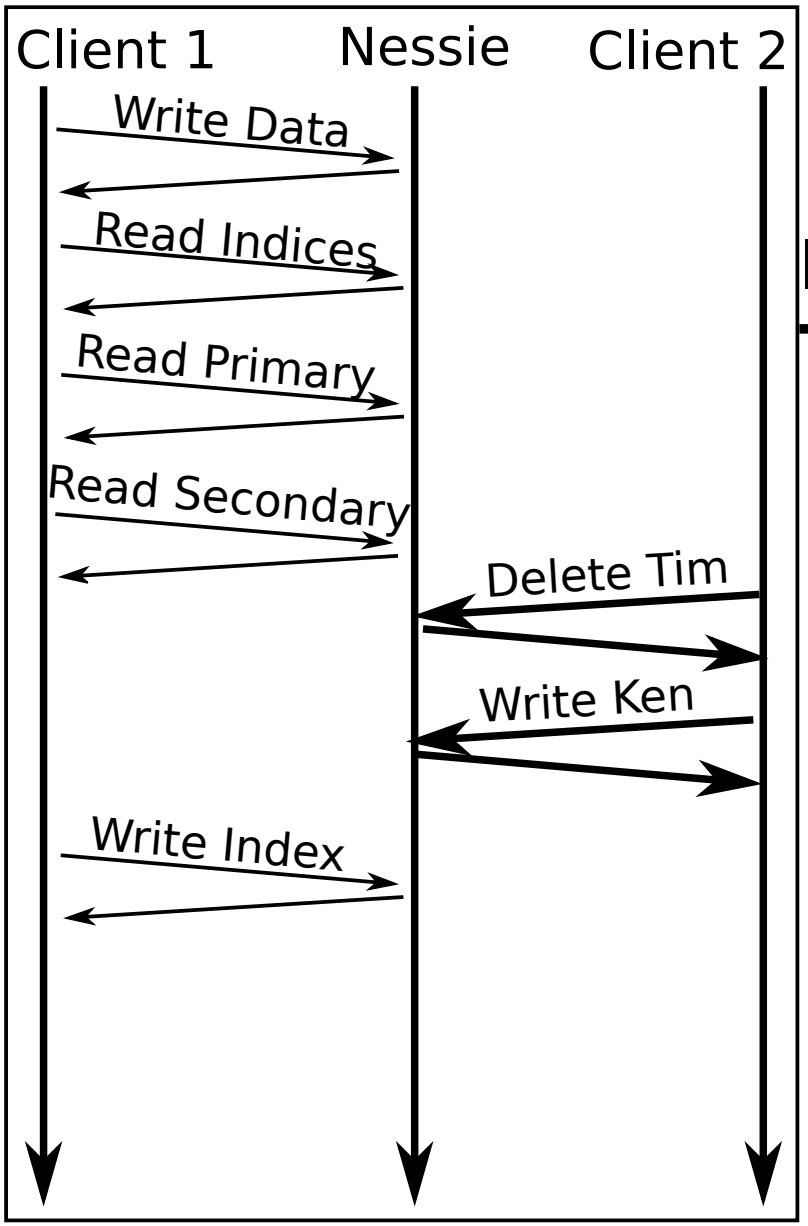
Index Table

10
6
1
~~5~~ ~~0~~ 2

Data Table

Key	Value	Valid	
Sam	Pizza	T	1
Ken	**Cake**	**T**	2
			3
			4
Tim	Carrot	T	5
Dan	Bacon	T	6
Tim	**Eggs**	**F**	7
			8
			9
Ben	Apple	T	10
			11
			12
			13
			14

Nessie: Write with Conflict



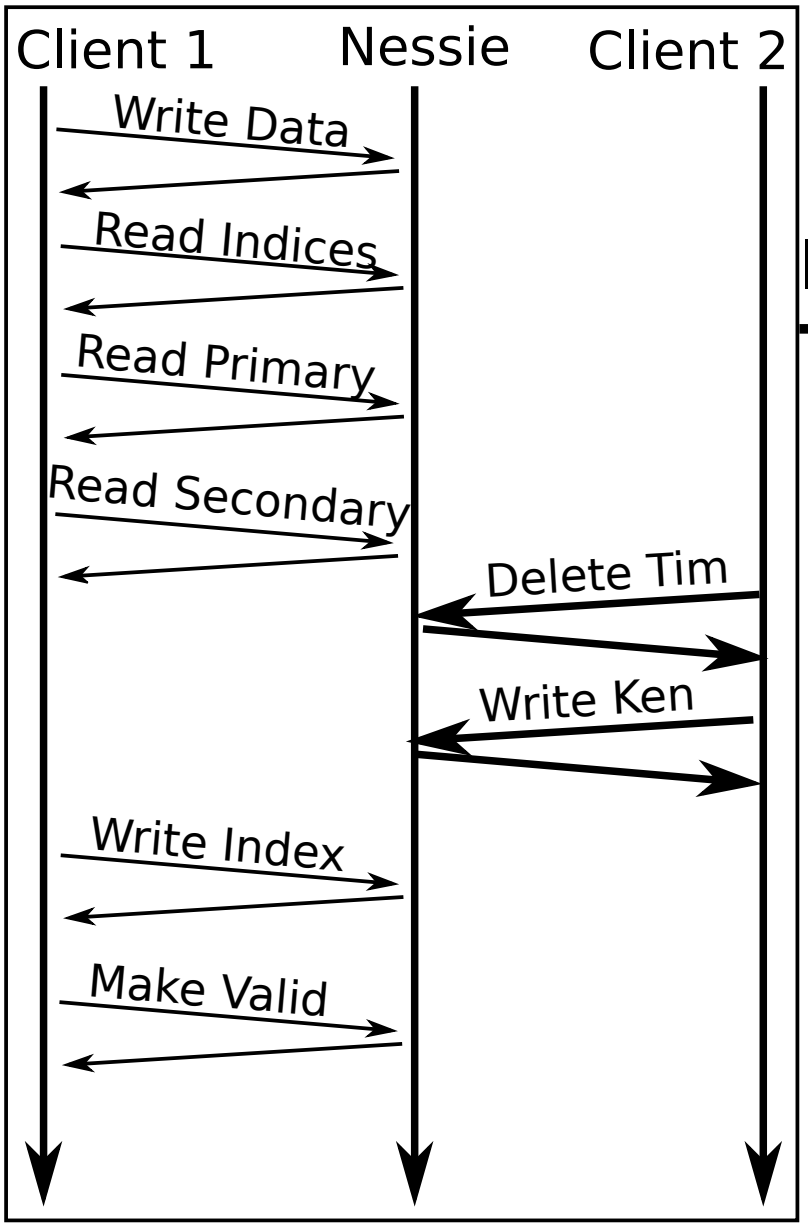
Index Table

10
6
1
~~5027~~

Data Table

	Key	Value	Valid
1	Sam	Pizza	T
2	**Ken**	**Cake**	**T**
3			
4			
5	Tim	Carrot	T
6	Dan	Bacon	T
7	**Tim**	**Eggs**	**F**
8			
9			
10	Ben	Apple	T
11			
12			
13			
14			

Nessie: Write with Conflict



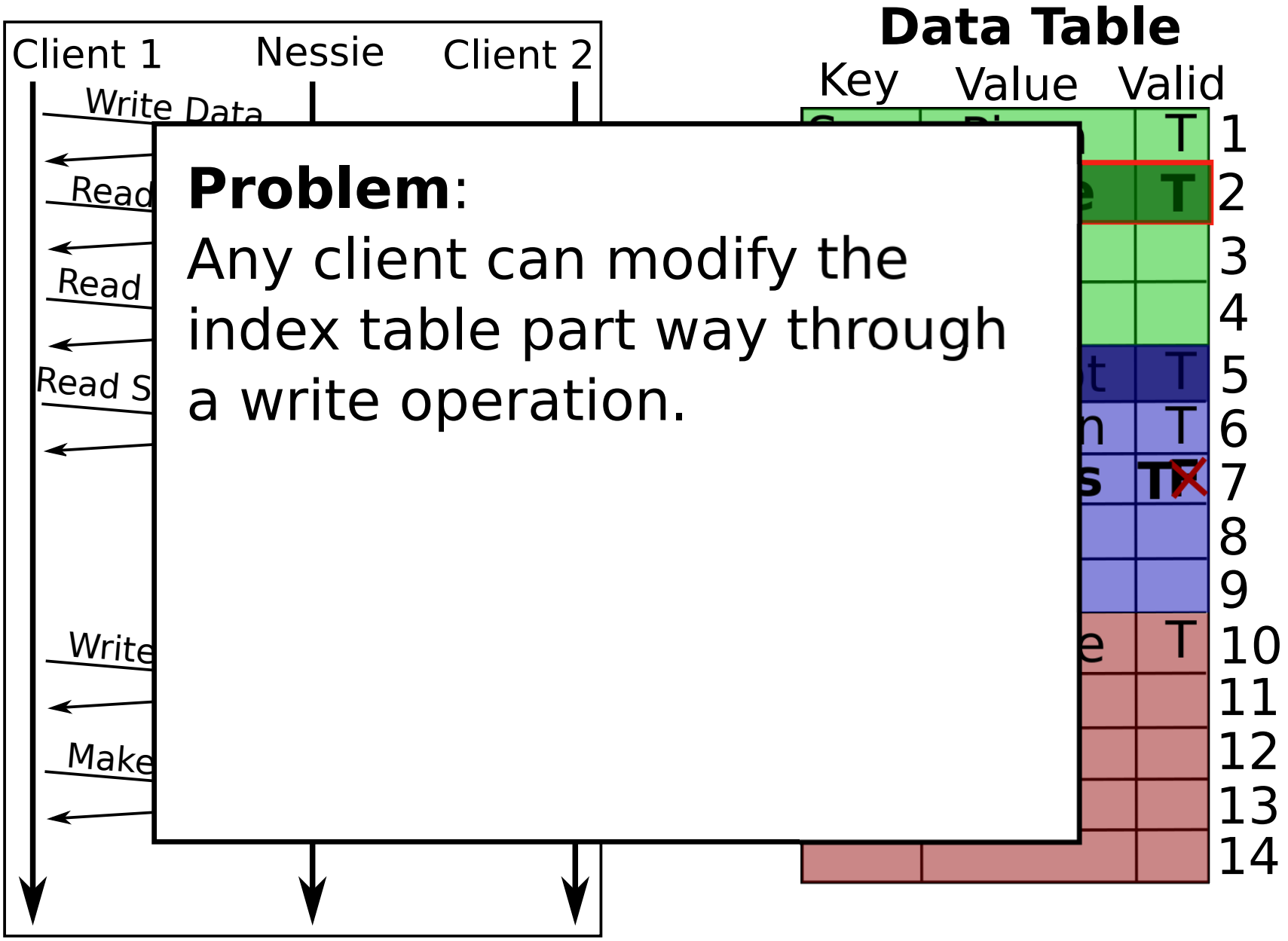
Index Table

10
6
1
~~5~~ ~~0~~ 27

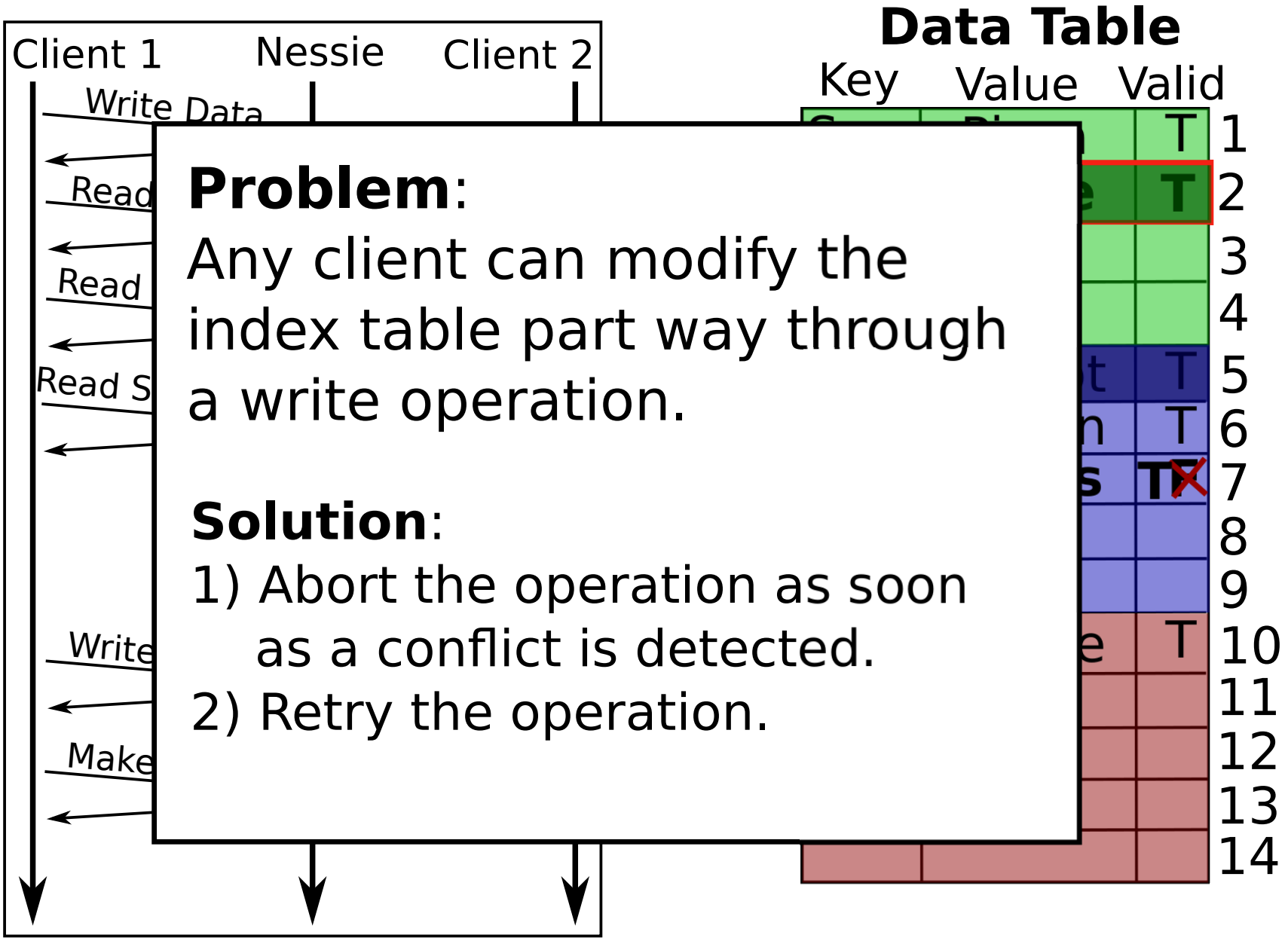
Data Table

Key	Value	Valid	
Sam	Pizza	T	1
Ken	**Cake**	**T**	2
			3
			4
Tim	Carrot	T	5
Dan	Bacon	T	6
Tim	**Eggs**	**T** ~~F~~	7
			8
			9
Ben	Apple	T	10
			11
			12
			13
			14

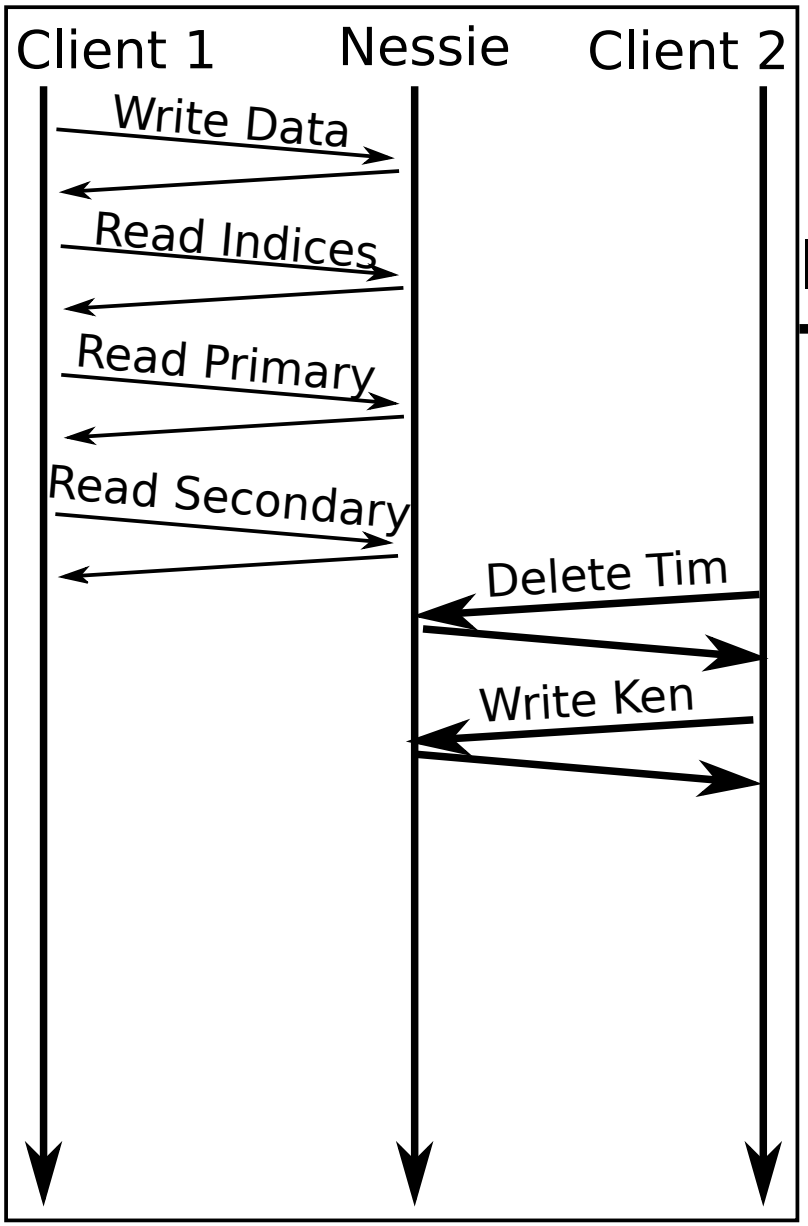
Nessie: Write with Conflict



Nessie: Write with Conflict



Nessie: Handling Conflicts



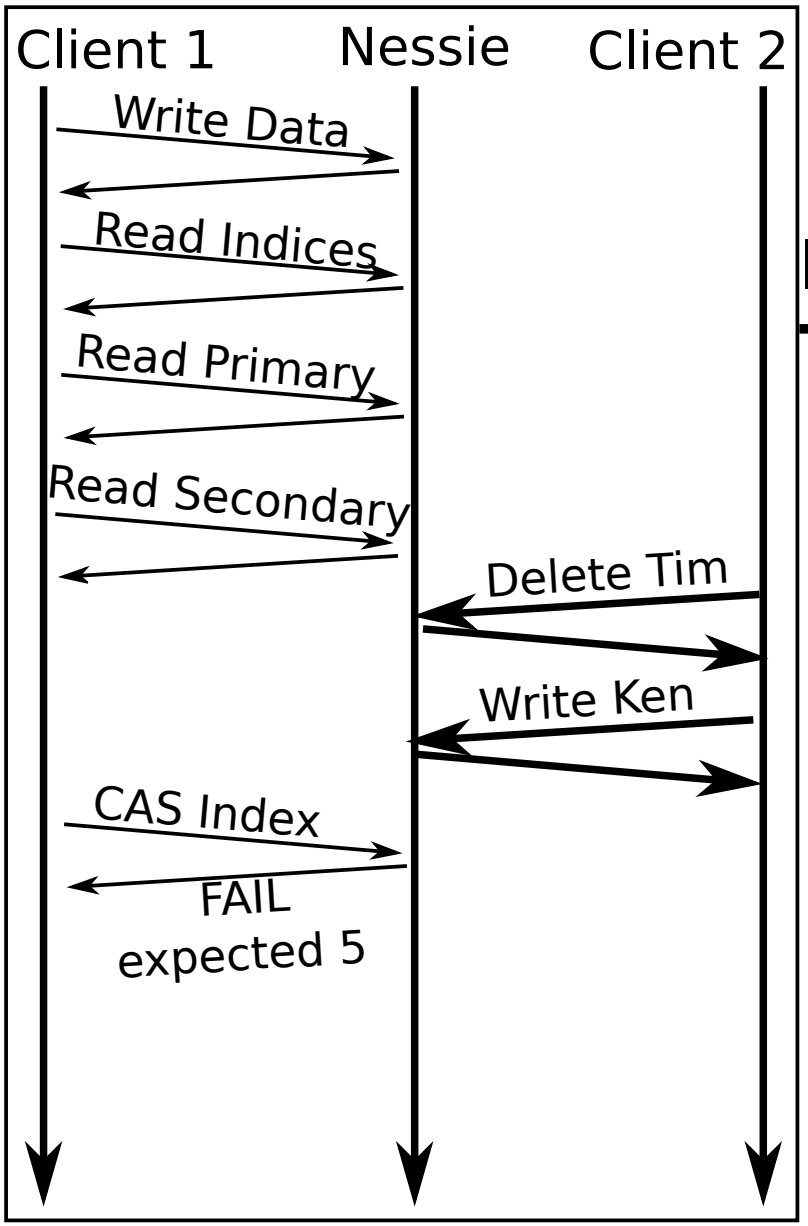
Index Table

10
6
1
~~5~~ 2

Data Table

Key	Value	Valid	
Sam	Pizza	T	1
Ken	**Cake**	**T**	2
			3
			4
Tim	Carrot	T	5
Dan	Bacon	T	6
Tim	**Eggs**	**F**	7
			8
			9
Ben	Apple	T	10
			11
			12
			13
			14

Nessie: Handling Conflicts



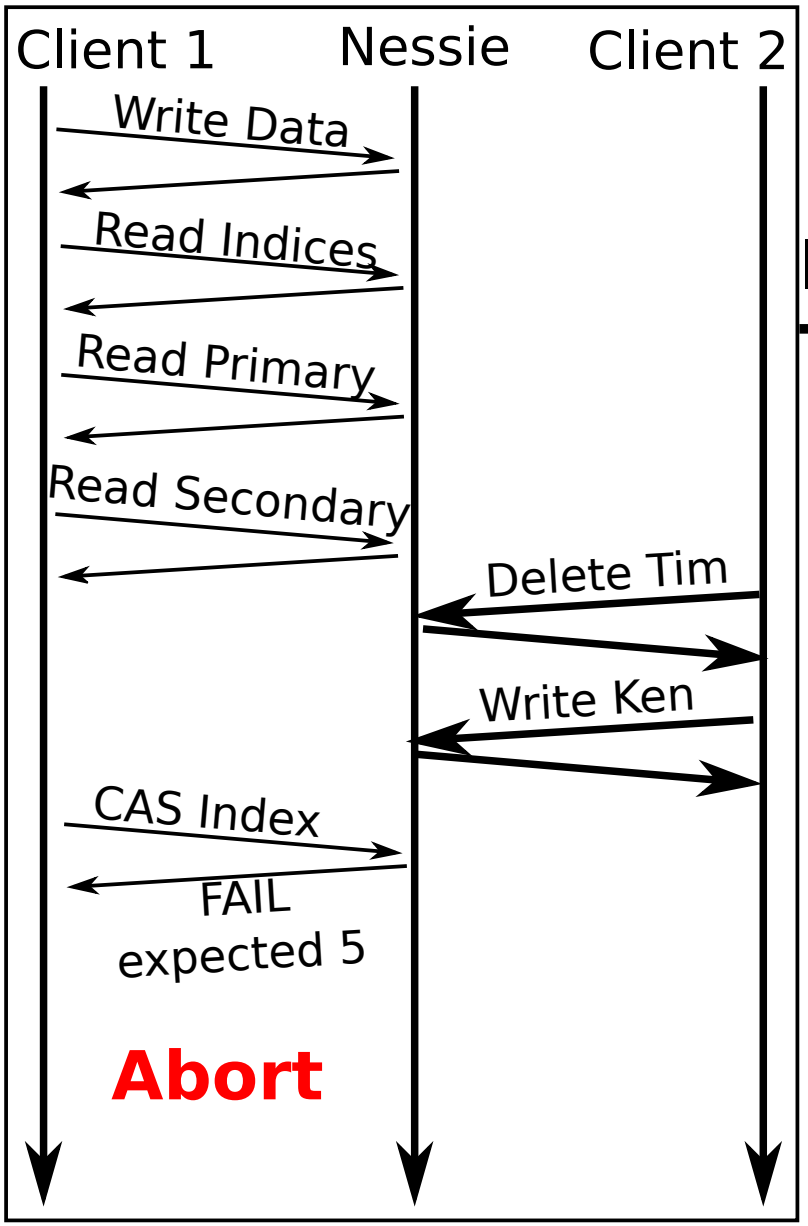
Index Table

10
6
1
~~5~~ 2

Data Table

Key	Value	Valid	
Sam	Pizza	T	1
Ken	**Cake**	**T**	2
			3
			4
Tim	Carrot	T	5
Dan	Bacon	T	6
Tim	**Eggs**	**F**	7
			8
			9
Ben	Apple	T	10
			11
			12
			13
			14

Nessie: Handling Conflicts



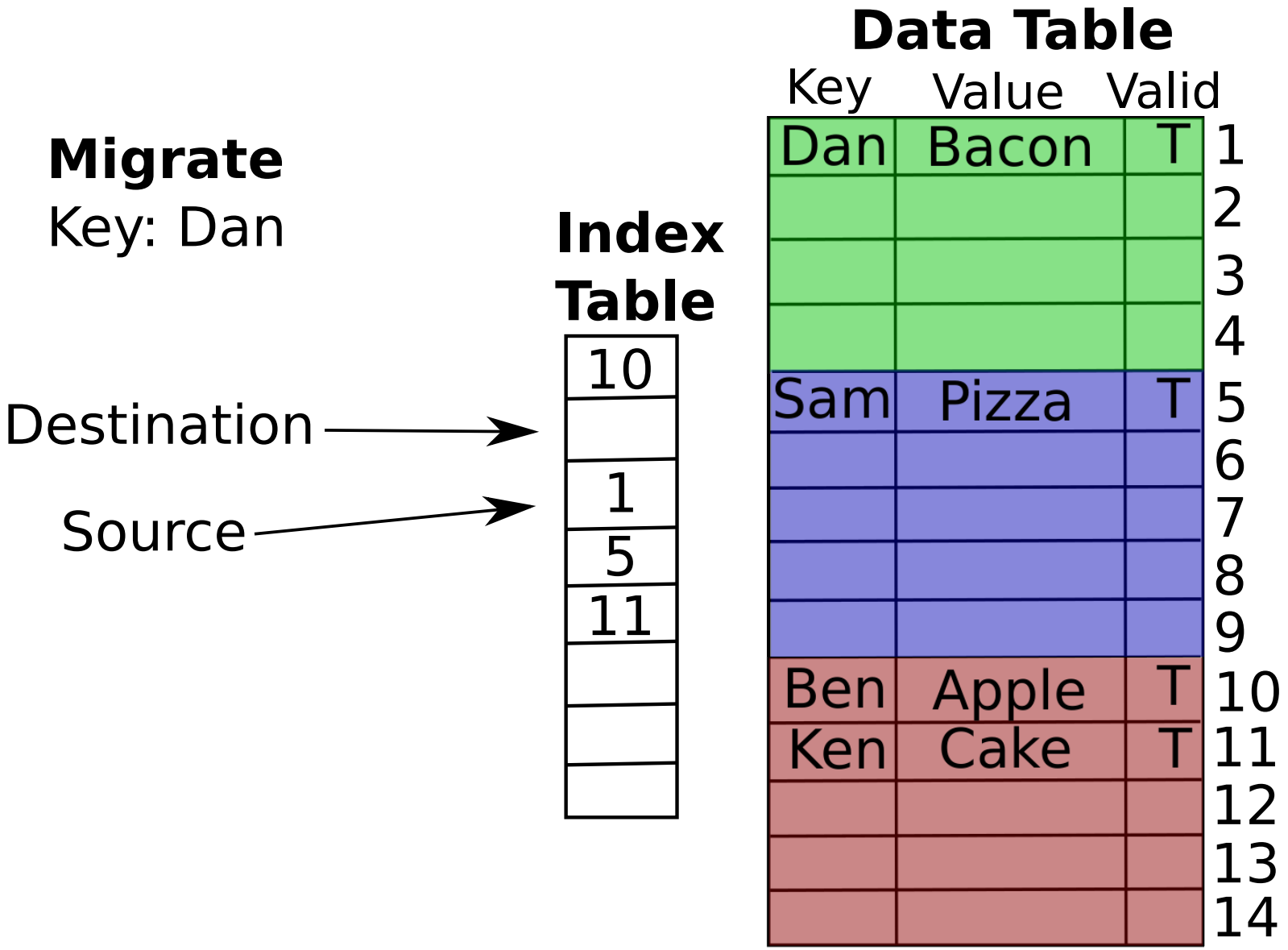
Index Table

10
6
1
~~5~~ 2

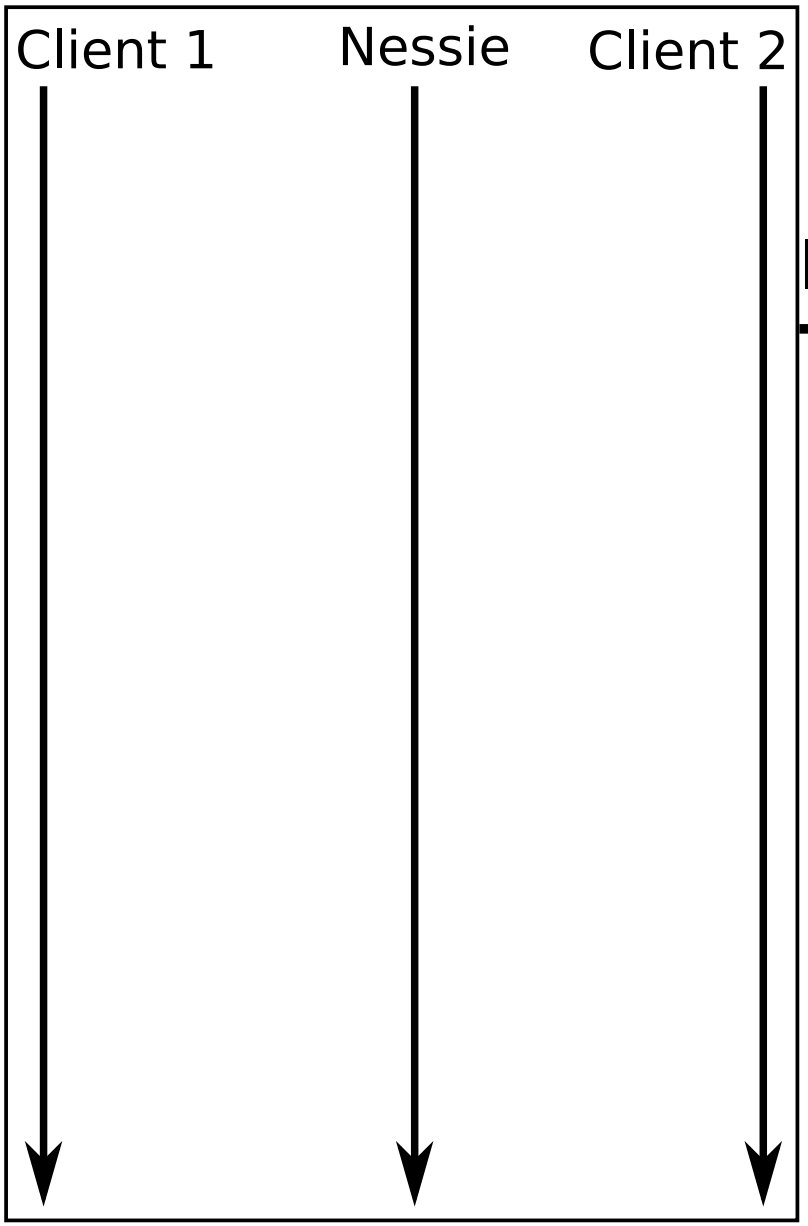
Data Table

Key	Value	Valid	
Sam	Pizza	T	1
Ken	**Cake**	**T**	2
			3
			4
Tim	Carrot	T	5
Dan	Bacon	T	6
Tim	**Eggs**	**F**	7
			8
			9
Ben	Apple	T	10
			11
			12
			13
			14

Nessie: Migration



Nessie: Migration



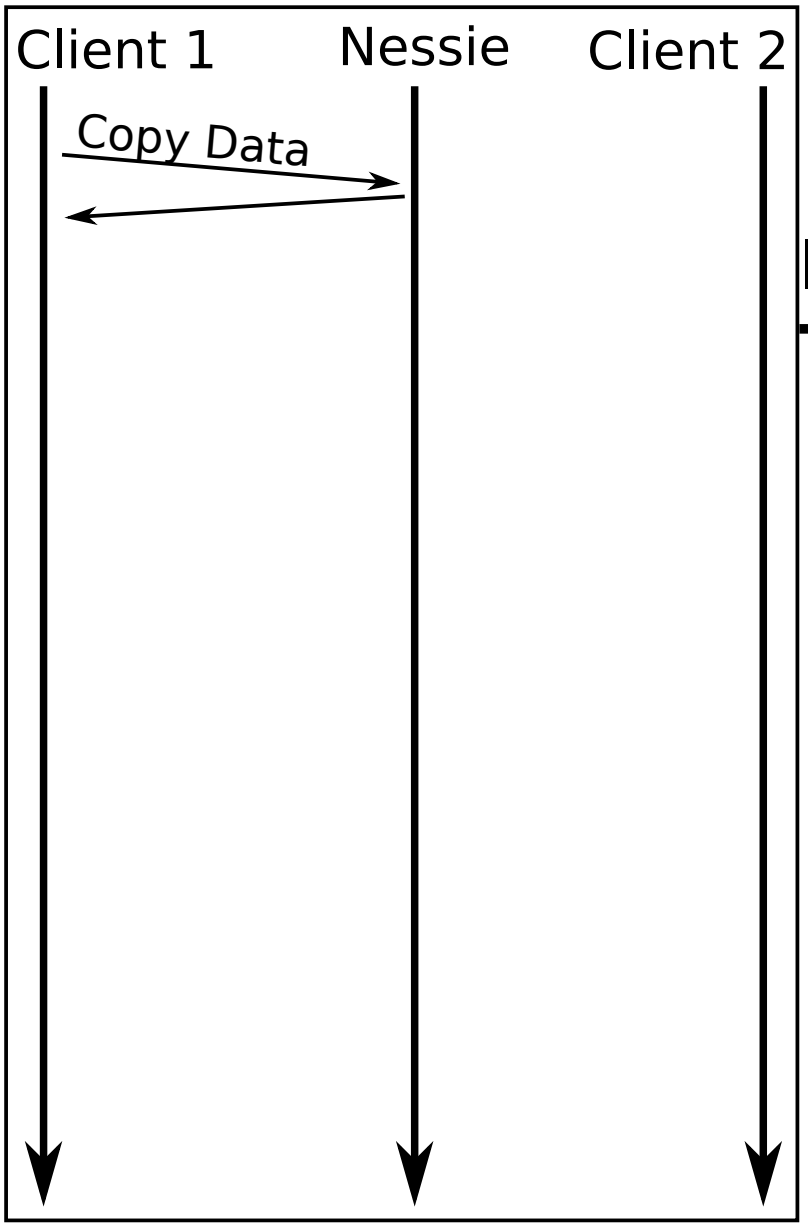
Index Table

10
1
5
11

Data Table

	Key	Value	Valid
1	Dan	Bacon	T
2			
3			
4			
5	Sam	Pizza	T
6			
7			
8			
9			
10	Ben	Apple	T
11	Ken	Cake	T
12			
13			
14			

Nessie: Migration



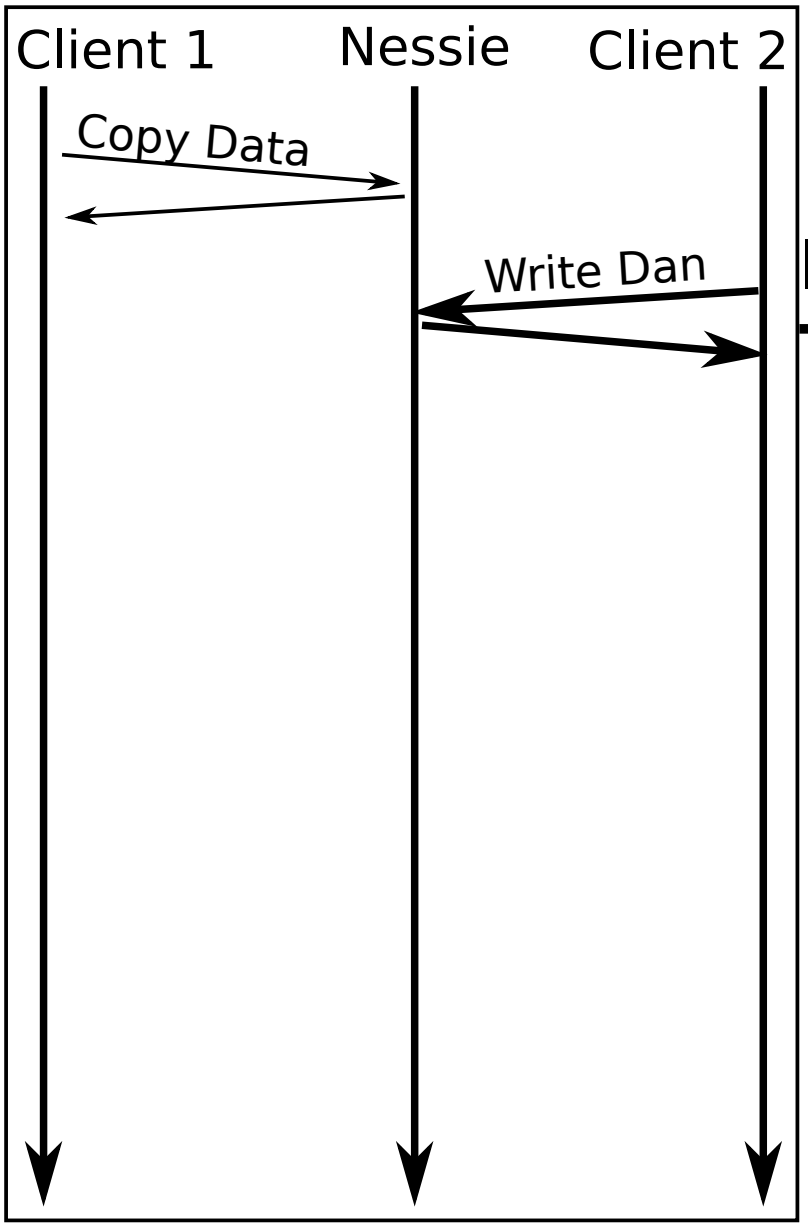
Index Table

10
1
5
11

Data Table

Key	Value	Valid	
Dan	Bacon	T	1
			2
			3
			4
Sam	Pizza	T	5
Dan	**Bacon**	**F**	6
			7
			8
			9
Ben	Apple	T	10
Ken	Cake	T	11
			12
			13
			14

Nessie: Migration



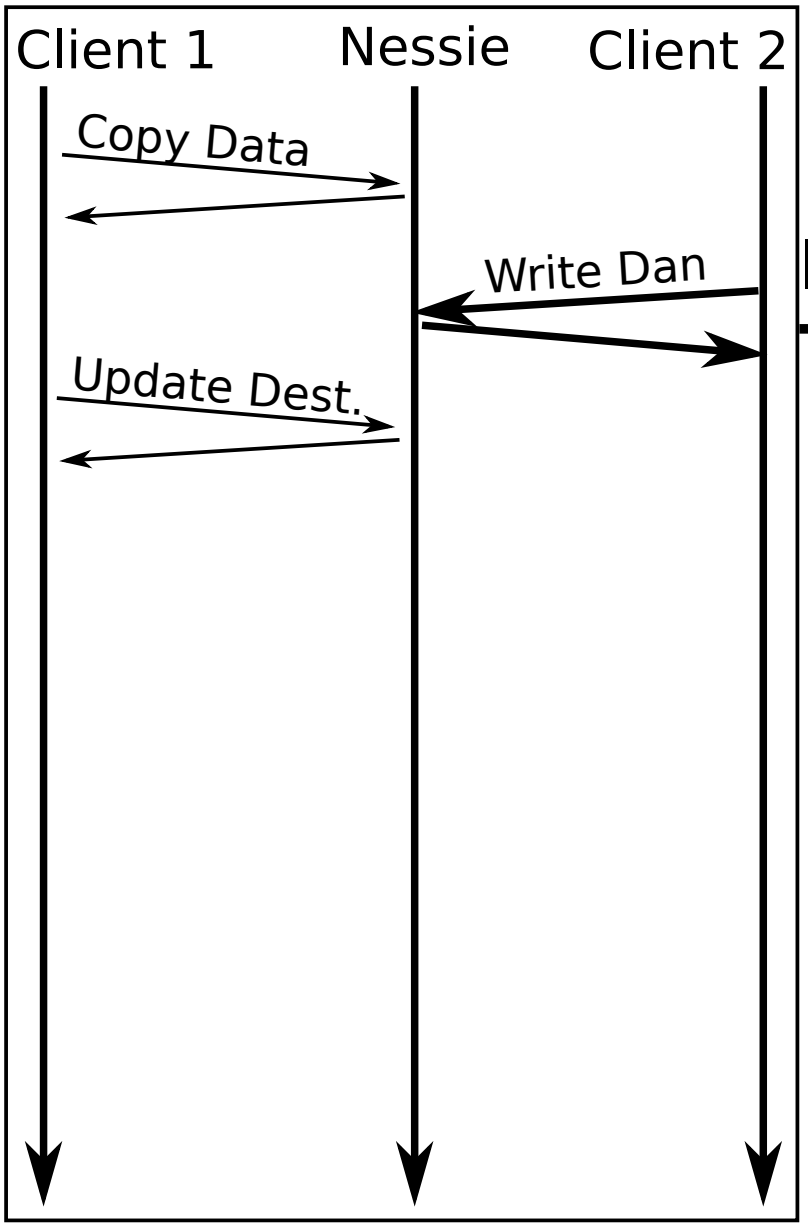
Index Table

10
~~1~~2
5
11

Data Table

Key	Value	Valid	
Dan	Bacon	T	1
Dan	**Tofu**	**T**	2
			3
			4
Sam	Pizza	T	5
Dan	**Bacon**	**F**	6
			7
			8
			9
Ben	Apple	T	10
Ken	Cake	T	11
			12
			13
			14

Nessie: Migration



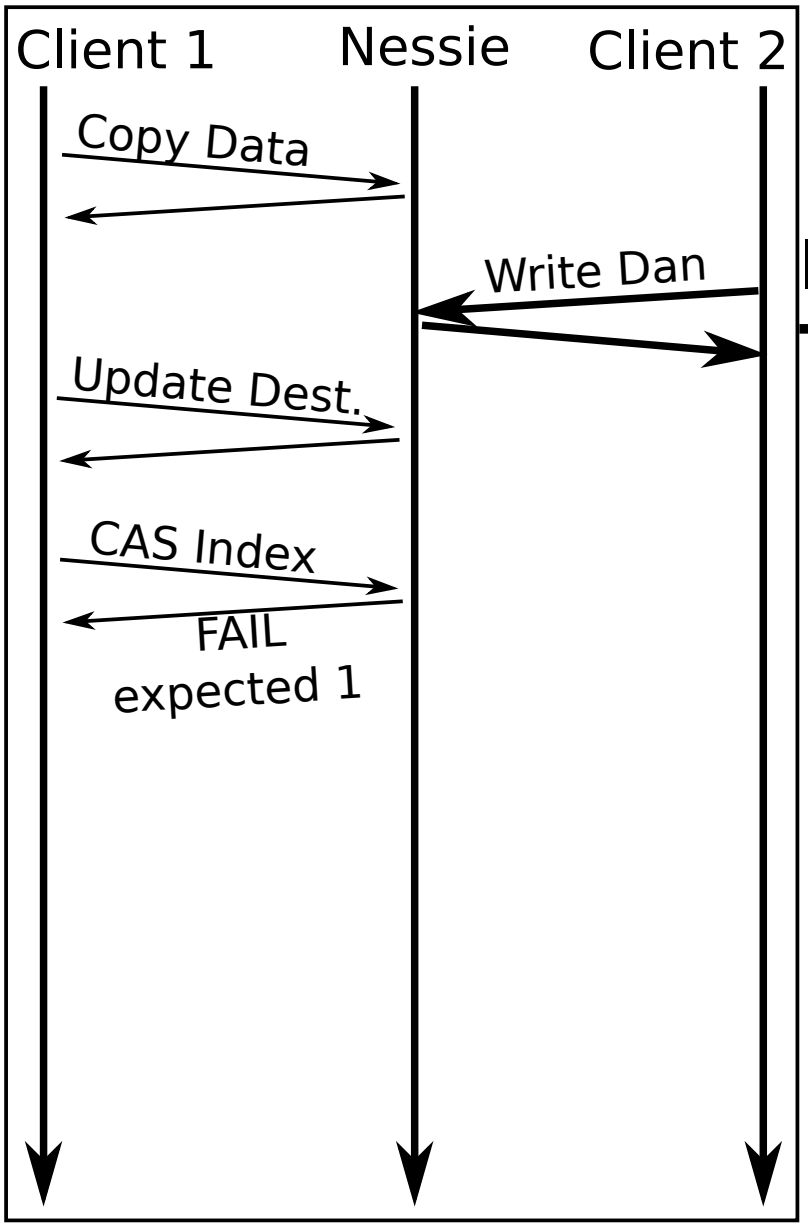
Index Table

10
6
~~1~~2
5
11

Data Table

Key	Value	Valid	
Dan	Bacon	T	1
Dan	**Tofu**	**T**	2
			3
			4
Sam	Pizza	T	5
Dan	**Bacon**	**F**	6
			7
			8
			9
Ben	Apple	T	10
Ken	Cake	T	11
			12
			13
			14

Nessie: Migration



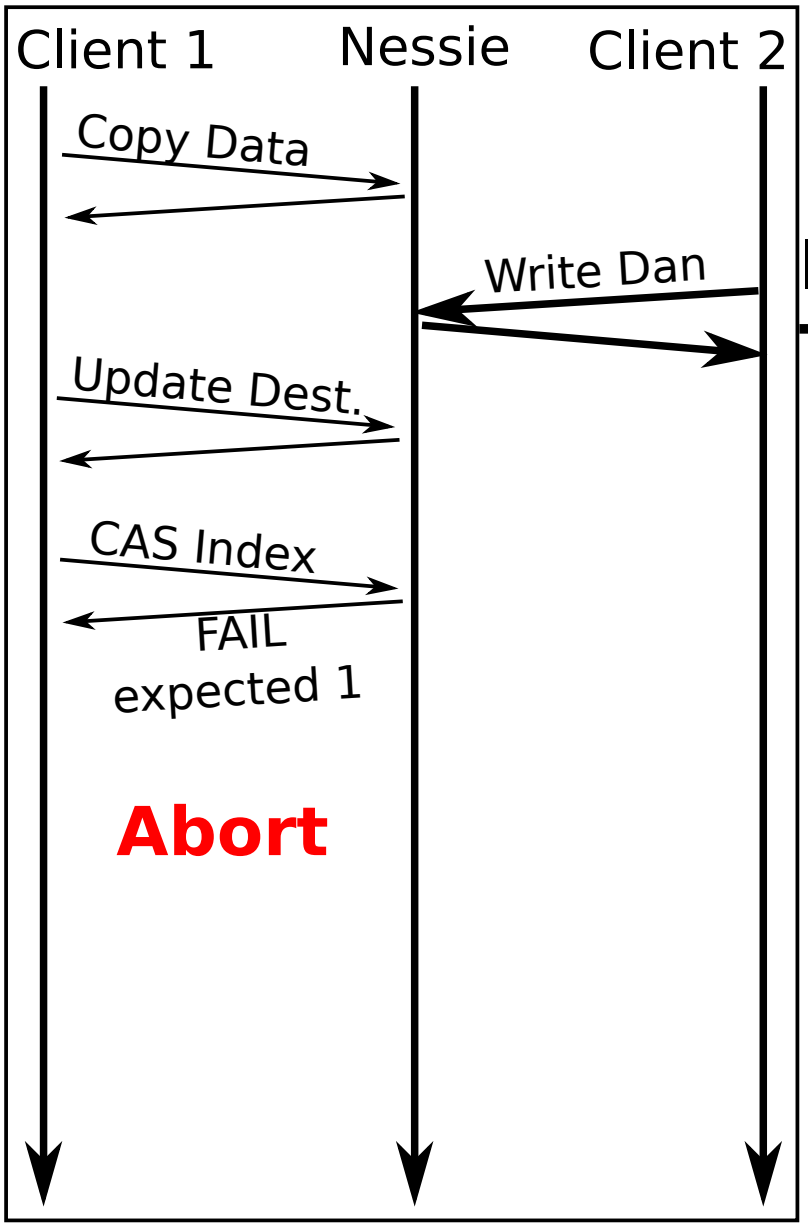
Index Table

10
6
~~1~~2
5
11

Data Table

Key	Value	Valid	
Dan	Bacon	T	1
Dan	**Tofu**	**T**	2
			3
			4
Sam	Pizza	T	5
Dan	**Bacon**	**F**	6
			7
			8
			9
Ben	Apple	T	10
Ken	Cake	T	11
			12
			13
			14

Nessie: Migration



Index Table

10
6
~~1~~2
5
11

Data Table

Key	Value	Valid	
Dan	Bacon	T	1
Dan	**Tofu**	**T**	2
			3
			4
Sam	Pizza	T	5
Dan	**Bacon**	**F**	6
			7
			8
			9
Ben	Apple	T	10
Ken	Cake	T	11
			12
			13
			14

Performance Tradeoffs

Expected number of roundtrips for a write operation in Nessie:

Load Factor	Roundtrips
0.10	2.5
0.25	3.5
0.50	5.8
0.75	9.9
0.90	13.3

Nessie's Design Complexity

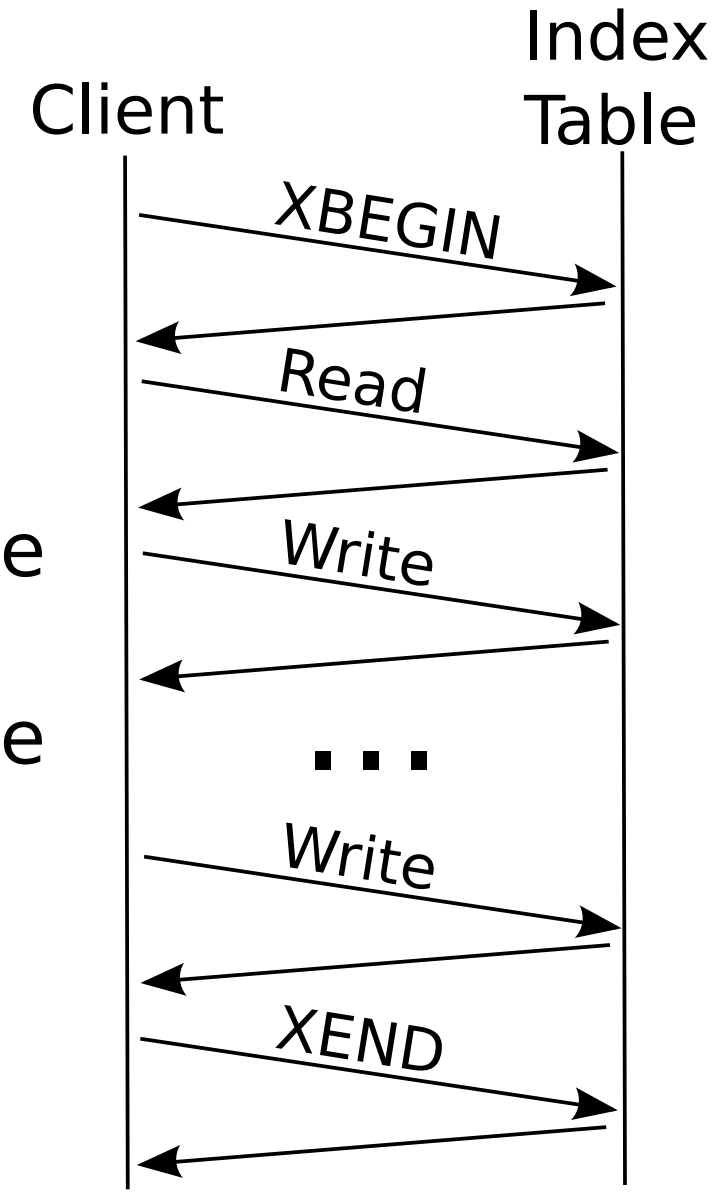
- Contention over the index table creates complexity
- Our solution mirrors that of Hardware Transactional Memory
  - Opimistically attempt operation
  - Check for conflicting operations
  - Abort and retry if a conflict is detected
- NIC support of HTM would greatly improve Nessie
  - NIC handles detection and reporting of conflicts
  - Fewer roundtrips

Hardware Transactional Memory

```
XBEGIN
    Read Index Table
    Write to Index Table
    ...
    Write to Index Table
XEND
```

HTM + RDMA

```
XBEGIN
    Read Index Table
    Write to Index Table
    ...
    Write to Index Table
XEND
```



Summary

- Low latency is critical for in-memory distributed systems
- Nessie:
  - Exclusively RDMA reads, writes, and CAS operations
  - Achieves low-latency
  - Shared lock-free data
- HTM in conjunction with RDMA can greatly simplify Nessie
- We are excited to build Nessie and explore HTM