PREM KRISHNA CHETTRI

Distributed System Assignment 4 Submission Date: 11th Dec '15

Solution 1>

i) Partition separates X and Y from Z when all data items have version v0 say: XYZ

A= 100 (vo) A= 100(vo) A= 100(vo)

B= 100(vo) B= 100(vo) B= 100(vo)

A client reads the value of A and then writes it to B:

read quorum = 1+1 for A and B - client Reads A from X or Y

write quorum = 1+1 for B client Writes B at X and Y

ii) Client can access only server Z: read quorum = 1, so client cannot read, write quorum = 1 so client cannot write, therefore neither operation takes place.

iii) After the partition is repaired, the values of A and B at Z may be out of date, due to clients having written new values at servers X and Y. e.g. versions v1:

XYZ
A= 200(v1) A= 200(v1) A= 100(vo) B= 300(v1) B= 300(v1) B= 100(vo)

Then another partition occurs so that X and Z are separated from Y.

The client Read request causes an attempt to obtain a read quorum from X and Z. This notes that the versions (v0) at Z are out of date and then Z gets up-to-date versions of A and B from X.

Now the read quorum = 1+1 and the read operation can be done. Similarly the write operation can be done.

Solution 2.

In the two-phase commit protocol: the 'uncertain' period occurs because a worker has voted *yes* but has not yet been told the outcome. (It can no longer abort unilaterally).

In the three-phase commit protocol: the workers 'uncertain' period lasts from when the worker votes *yes* until it receives the pre-commit request. At this stage, no other participant can have committed. Therefore if a group of workers discover that they are all 'uncertain' and the coordinator cannot be contacted, they can decide unilaterally to abort.

Solution 3.

Virtual nodes are similar to the single physical node in the system but capable of handling the responsibility of other nodes during the time of failure of its neighbors. Virtual nodes have an advantage over physical nodes during failure scenario. If a node becomes unavailable, the load balances across other remaining nodes. It will transfer back the responsibility of the node, once it will be available again. Also, we can dynamically configure our virtual node, based on the capacity of the node.

Solution 4.

Dynamo uses vector clocking (list of object, counter) for determining the causality of the two versions of the each object. For, causally related version, we accept the final version by examining the vector clock of each of the versions. If the counter in the first object is less then or equal to all the objects in the second clock, then first is an ancestor of second, on which case, first is the older copy and can be forgotten.

Solution 5.

N : Performance (number of nodes marks as the performance factor)

R : Availability (read marks as the availability)

W : Durability (write marks as the durability)

Dynamo uses quorum-based approach in order to avoid conflicts. In this approach we intend to have high availability. We represent R as read (Availability), W as write (Durability) and N is the total number of nodes in the system (performance). So, for any system to be consistent, the factor R + W must be greater then N, making sure that there is atleast one node which overlaps between a read and write operation in a system.

In Dynamo, it's configurable as it depends on the need of an application, so if a system need high availability (fast reading), we can make R = N and W = 1, so any node can be read, whereas if we have W = N and R =1, which means (high durable) fast writing. The typical configuration of dynamo is W = R = 2 when N = 3.

Solution 6.

Sloppy quorum is a variation of the quorum based approach in which all read and write operation are performed on the first N healthy node of the preference list, which may not necessary be the first N node encountered in the system. This provides, availability on the temporary failure and partition of the node. As dynamo, sacrifices, consistency over availability, it helps to maintain the higher availability of the system.

Solution 7.

a>

$h_i(x) = ((x^2 + x^3) * i) \bmod m,$

h1(2013) = ((2013 * 2013 + 2013*2013 *2013) * 1) mod 32 = 14

h2(2013) = ((2013 * 2013 + 2013*2013*2013) * 2) mod 32 = 28

h3(2013) = ((2013 * 2013 + 2013*2013*2013) * 3) mod 32 = 10

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
										1				1														1			

h1(2010) = ((2010 * 2010+ 2010*2010 *2010) * 1) mod 32 = 12

h2(2010) = ((2010 * 2010 + 2010*2010*2010) * 2) mod 32 = 24

h3(2010) = ((2010 * 2010 + 2010*2010*2010) * 3) mod 32 = 4

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
				1						1		1		1										1				1			

h1(2007) = 24

h2(2007) = 16

h3(2007) = 8

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
				1				1		1		1		1		1								1				1			

h1(2004) = 16

h2(2004) = 0

h3(2004) = 16

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1				1				1		1		1		1		1								1				1			

h1(2001) = 18

h2(2001) = 4

h3(2001) = 22

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1

1				1				1		1		1		1		1		1				1		1				1			

h1(1998) = 28

h2(1998) = 24

h3(1998) = 20

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1				1				1		1		1		1		1		1		1		1		1				1			

b>

From the above table, we can see that 12, 14, 2300 has been hashed into but we never encountered any of these while selecting the numbers.

Solution 8.

a>

1> Single server : - Yes as it is in different rack.
2> Single rack :- Yes it is fault tolerant.
3> Single Data Center :- No as DC octet, 231 is the common to all.

b>

1> Single server : - Yes it is fault tolerant.
1> Single rack :- Yes it is fault tolerant.
2> Single Data Center :- Yes it is fault tolerant.

c>

2> Single server : - Yes it is fault tolerant.
3> Single rack :- No, all are in same rack.
4> Single Data Center :- No as DC octet, 231 is the common to all.

d>

1> Single server : - Yes it is fault tolerant.
2> Single rack :- Yes it is fault tolerant.
3> Single Data Center :- No as DC octet, 231 is the common to all.

e>

1> Single server : - Yes it is fault tolerant, two servers 4 and 5
2> Single rack :- No, all are in rack 3.
3> Single Data Center :- No as DC octet, 2 is the common to all.