BIGDATA – Assignment 3:

Replication, balancing and fault tolerance in HDFS

Contributors:

Suchi Siwach
Muthupriya Shankaran
Augustin Ionut Ionascu

DataCenter/Cluster configuration:
Number of racks: 3
Number of DataNodes: 15 (5 per rack)
Hard drives per DataNode: 4 (2 TB each)
Default block size: 128 Mb
Default replication rate: 3

Step 1: Divide the files into blocks

Files in the Hadoop ecosystem are divided into blocks, which typically have 128 Mb each. The last block will contain less data if the file size doesn't divide exactly to 128.

	Filename	File size	Blocks
File 1	**Cities.json**	**350 Mb**	A(128Mb), B(128Mb), C(94Mb)
File 2	**Citizens.json**	**620 Mb**	D(128Mb), E(128Mb), F(128Mb), G(128Mb), H(108Mb)
File 3	**Companies.json**	**500 Mb**	I(128Mb), J(128Mb), K(128Mb), L(116Mb)

Step 2: Load the files into the DataCenter below

The assignment of replicas is considering the following rules, in this case, when the default replication factor is 3:

- one replica is placed on local / random node
- both second and third replicas are being assigned to another rack, but to different nodes

It is also taken into consideration load balancing across data nodes (in terms of number of blocks). The size of data per rack however, would be slightly different because blocks C, H and L differ in size.

DataCenter	Rack 1	Rack 2	Rack 3
DataNodes	A1, F3, L2	A2, E1, J3	B2, F1, K3
	C2, G1, L3	A3, G2, K1	B3, H2, L1
	C3, I2	B1, G3	C1, H3
	D1, I3	D2, H1	E2, I1
	F2, J1	D3, J2	E3, K2

Step 3: Increase the replication rate for Citizens.json

The replicas assignment is following the same rules as at point 2 and in addition we also consider the following additional rules:

- maximum 2 replicas per rack if possible
- maximum one replica per node

DataCenter	Rack 1	Rack 2	Rack 3
DataNodes	A1, F3, L2, E5	A2, E1, J3, F5	B2, F1, K3, D5
	C2, G1, L3, H4	A3, G2, K1, I4	B3, H2, L1, G4
	C3, I2, B4, H5	B1, G3, C4, I5	C1, H3, A4, G5
	D1, I3, B5, K4	D2, H1, C5, L4	E2, I1, A5, J4
	F2, J1, E4, K5	D3, J2, F4, L5	E3, K2, D4, J5

Step 4a: Fail node(s) and rebalance the under-replicated blocks

Let's consider that the whole rack 2 is off. After the detection of failures of the data nodes in rack 2, the replication engine will identify the under replicated blocks by comparing the data nodes left in the cluster to the replication factor. Further, the engine will set up a replication queue, by the following principles:

- a block with one replica has the greatest priority;
- the lowest priority is assigned to block with a number of replicas to replication factor ratio greater than ⅔;

The nodes with the same priority level have been ordered randomly.

DataCenter	Rack 1	Rack 2	Rack 3
DataNodes	A1, F3, L2	**A2(1st), G3(6th), E1(11th)**	B2, F1, K3
	C2, G1, L3	**A3(2nd), J2(7th), H1(12th)**	B3, H2, L1
	C3, I2	**D2(3rd), J3(8th)**	C1, H3
	D1, I3	**D3(4th), K1(9th)**	E2, I1
	F2, J1	**G2(5th), B1(10th)**	E3, K2

Step 4b: Replication Priority Queue

Highest	A	D	G	J	K	B	E	H									Lowest