

ByteOrderedPartitioner

Byte Ordered Partitioner (BOP) is a scheme to organize how to place the keys in the Cassandra cluster node ring. Unlike the [RandomPartitioner](#) (RP), the raw byte array value of the row key is used to decide which nodes store the row. Depending on the distribution of the row keys, you may need to actively manage the tokens assigned to each node to maintain balance.

As an example, if all row keys are random (type 4) UUIDs, they are already evenly distributed. However they are 128 bits, unlike the 127 bit tokens used by RP, and the initial tokens must be specified as hex byte strings instead of decimal integers. Here is python code to generate the initial tokens, in a format suitable for cassandra.yaml and nodetool:

```
def get_cassandra_tokens_uuid4_keys_bop(node_count):
    # BOP expects tokens to be byte arrays, specified in hex
    return ["%032x" % (i*(2**128)/node_count)
            for i in xrange(0, node_count)]
```

Note that even if your application currently uses random UUID row keys for all data, you may run into balancing issues later on if you add new data with non-uniform keys, or keys of a different size. This is why RP is recommended for most applications.

<https://c.statcounter.com/9397521/0/fe557aad/1/> | stats